Mycobacterium tuberculosis aggregates affect the early macrophage response to infection and are detectable in human lung tissue

Hylton Errol Rodel

Division of Infection and Immunity

University College London

PhD Supervisors: Prof. Alex Sigal and Prof. Mahdad Noursadeghi

A thesis submitted for the degree of

Doctor of Philosophy

University College London

December 2022

Declaration

I, Hylton Errol Rodel, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Mycobacterium tuberculosis (Mtb) can infect macrophages as single or aggregated bacilli, where aggregate infection of macrophages was shown to have a substantially higher probability to result in macrophage death. Given that the response of macrophages to Mtb infection may determine the infection trajectory, it is important to understand the macrophage response to infection with Mtb aggregates. Here I investigated the early transcriptional response of monocyte derived macrophages (MDMs) to Mtb aggregate infection. I found that Mtb aggregates elicited the highest TNF-α and pro-inflammatory response relative to single Mtb bacilli. Additionally, aggregate-mediated MDM death was dependent on infection with live Mtb aggregates. I also investigated macrophage acidification in response to infection with Mtb aggregates and found that acidification, per Mtb bacillus, decreased as aggregate size increased. This suggests that Mtb aggregates have an advantage over single bacilli due to a weaker host response per mycobacterium. I also quantified Mtb aggregate number in human lung tissue sections using custom digital image analysis pipelines and developed a convolutional neural network (CNN) model, HyRoNet, to automate and expand the analysis. I found that Mtb aggregates occurred often, but not exclusively, in association with the granulomatous cavity surface. Together, these observations suggest a potentially important role for Mtb aggregation in the pathogenesis of Mtb.

Impact statement

Understanding the pathogenic mechanisms of *Mycobacterium tuberculosis* is an important step in addressing tuberculous disease. Investigating the physical parameters of an infection, such as bacterial aggregation state, and how these parameters affect the host response could have applications in research as well as translational medicine. I have shown that Mtb aggregates, which have been demonstrated to result in enhanced cell death for host cells, elicit different transcriptional responses in macrophages relative to infection with single or multiple singlet Mtb bacteria. Identifying such unique transcription patterns could result in the development of therapeutics targeting these pathways to potentially alter infection trajectory. Quantitative data on the size distribution of aggregates in the transmission of tuberculosis could inform decision-making on the necessary level of respiratory protection (N95 respirators, surgical masks, or cloth masks) for TB patients and contacts, which may be useful in resource-limited settings

While there is a large body of raw data on Mtb distribution in human lung tissues in the form of histological image databases, these remain largely untapped as visual examination requires a huge amount of expert labour. The development of tools to automate and quantify the data in these reserves is an important barrier to overcome in order to gain access to new insights. The recent surge in interest for machine learning techniques highlights a potential toolset to solve this problem. I have developed a custom CNN model targeting the quantification of Mtb bacilli in such image databases. The information gleaned from the application of such tools to large datasets could inform on disease progression and be applied in a clinical context for fast and automated medical image analysis.

Acknowledgements

First and foremost, I would like to thank my supervisors for their support during the duration of the PhD. I would like to thank Professor Alex Sigal for encouraging me to pursue my interest in computational biology and bioinformatics. For sending me to conferences, seminars, and workshops to hone my skillset and for supporting my professional development. I would also like to thank Professor Mahdad Noursadeghi, at UCL, for being a font of knowledge and a constant source of inspiration to me. I would also like to thank the members of my thesis committee, Prof Benni Chain and Al Leslie for always having the right blend of constructive criticism and motivation to spur me forward. I'd like to thank all members of Sigal lab, past and present, with a special mention to Jessica Railton, Ana Moyano de Las Muelas, Mallory Bernstein, Shi-Hsia Hwa, Yashica Ganga, Isabella Markham Ferreira, Gil Lustig and Deeqa Mahamed. The support of these amazing people, at a professional and personal level, has proved nothing short of invaluable and I can't imagine a finer or more capable group of scientists. I'd also like to thank Dr Steyn and his lab for all the help with the tissue slides. To the AHRI technical staff, in particular Devin Murugan, Pamela Ramkalawon and Hollis Shen, thank you for always helping and tolerating my innumerable requests. I'd also like to thank my family and friends. Mom and Dad for loving and supporting me above and beyond any reasonable expectations and my sister for blindly supporting me regardless of my actual capabilities. I'd also like to take this opportunity to thank my family by marriage who have all likewise supported me through all the trials and tribulations of the PhD. To my friends, among whose number overlap with my work colleagues, I'd like to say thank you for always being there when I needed to bend an ear, when I needed distraction or when I needed to take a second to smile at something silly. Also, a special shout out to Muffin, my dog, who I only ever get to visit back home on the odd occasion. Finally, and most importantly, I'd like to thank Rozlyn. My life partner, my wife, and my best friend. You'll have my heart and adoration until I'm nought but memory. Without you, I'd not have gotten far at all.

Funding

I would like to thank the AHRI-UCL PhD studentship program for funding my studies through University College London. As well as the Bill & Melinda Gates Foundation (Award OPP1116944) for funding my research.

Table of contents

List of figures

List of tables

Abbreviations

TB - tuberculosis

Mtb – *Mycobacterium tuberculosis*

NTM – Non-tuberculous mycobacteria

ZN – Ziehl-Neelsen

MDM – Monocyte derived macrophage

PAMP – pathogen associated molecular pattern

DAMP – damage associated molecular pattern

TLR – Toll like receptor

TNF-α – tumour necrosis factor alpha

TNFR – Tumour necrosis factor receptor

DE – Differential expression

PI3P - phosphatidylinositol 3-phosphate

CNN – Convolutional neural network

PtpA - protein tyrosine phosphatase

PCA – principal component analysis

DE – Differential expression

ML – machine learning

R - Red

G - Green

B - Blue

C - Cyan

M - Magenta

Y - Yellow

ZN-Mtb – ZN stained Mtb

AUC – area under curve

ROC – receiver operating characteristic

SSPN - sensitivity, specificity, positive predictive value and negative predictive value

PPV – Positive predictive value

NPV – Negative predictive value

RBC – Red blood cell

GSEA – Gene set enrichment analysis

α – gradient descent step size

λ – regularization parameter

Publications

Part of the work presented in this thesis has been presented in the following publication:

Rodel, H.E., Ferreira, I.A., Ziegler, C.G., Ganga, Y., Bernstein, M., Hwa, S.H., Nargan, K., Lustig, G., Kaplan, G., Noursadeghi, M. and Shalek, A.K., 2021. Aggregated Mycobacterium tuberculosis enhances the inflammatory response. Frontiers in microbiology, 12.

1 Chapter 1. Introduction

1.1 Mycobacterium tuberculosis

1.1.1 Tuberculosis disease

Tuberculosis (TB) disease affects millions of people each year [1]. Cardinal symptoms for active disease include fever, night sweats, coughing (with bloodied expectorate) and fatigue [1]. The end of 2021 saw a World Health Organization (WHO) tuberculosis report indicating a drastic decrease in TB case reporting and a concerning increase in the number of deaths caused by TB for the for the first time since 2005 [2]. This is likely due to a reduction in access to treatment, mediated by the Covid-19 pandemic [1, 2]. Most cases of TB occur in low-income countries, listed as high TB burden countries by the World Health Organization (WHO), with South Africa among them. In South Africa, the WHO's best estimate for total TB burden was 360 000 individuals, with a total mortality of around 58 000. This translates to the deaths of approximately 0.1% of the population of South Africa through infection with the Mtb bacteria in 2021 [2]. However, death and disease are not the only outcome for infection by *Mycobacterium tuberculosis* (Mtb) mycobacterium, the causative agent of TB. Most people who are infected will not progress to active tuberculous disease and can remain in a state known as latent TB infection. According to estimates using mathematical modelling approaches, the global burden of latent TB was approximately 23% of the global population in 2016. This translated to around 1.9 billion individuals [3]. This highlights one of the key challenges of the TB pandemic; understanding the transition to active disease, either from latent infection or as a result of host/pathogen interactions during primary infection events.

1.1.2 The Mycobacterium tuberculosis mycobacterium

Mtb is a facultative intracellular pathogen that was first characterized as the causative agent of tuberculosis by Robert Koch in 1882 [4]. The mycobacterium is part of a small group of closely related *Mycobacterium* species that can cause disease in human hosts, known as the *Mycobacterium* *tuberculosis* complex, which includes *M. africanum, M. microti and M. bovis* [5, 6]. Mtb is an ancient and highly successful human pathogen and phylogenetic studies have shown extreme intraspecies genetic homogeneity, suggesting that Mtb has clonally co-evolved with its human host from 20 000 – 35 000 years ago [6-8]. Other members of this genus are environmental organisms that do not cause tuberculosis and are known as non-tuberculous mycobacteria (NTM), although reports of NTM-associated disease have contributed to their clinical relevance in recent years [9-11]. A feature of mycobacteria is the presence of a unique cell envelope that contains mycolic acids. A property for which the genus is named.

Mycobacteria are characterized by a lipid rich cell envelope that is resistant to Gram staining and acid alcohol washing. The mycobacterial cell envelope is approximately 60% lipid in composition and is a 3-part structure consisting of an inner membrane, a cell wall and a capsule [12, 13] (**Figure 1.1 - adapted from Chen *et. al* [14]**). In Mtb, the bacterial capsule is composed predominantly of neutral polysaccharides and proteins, including α-glucans,

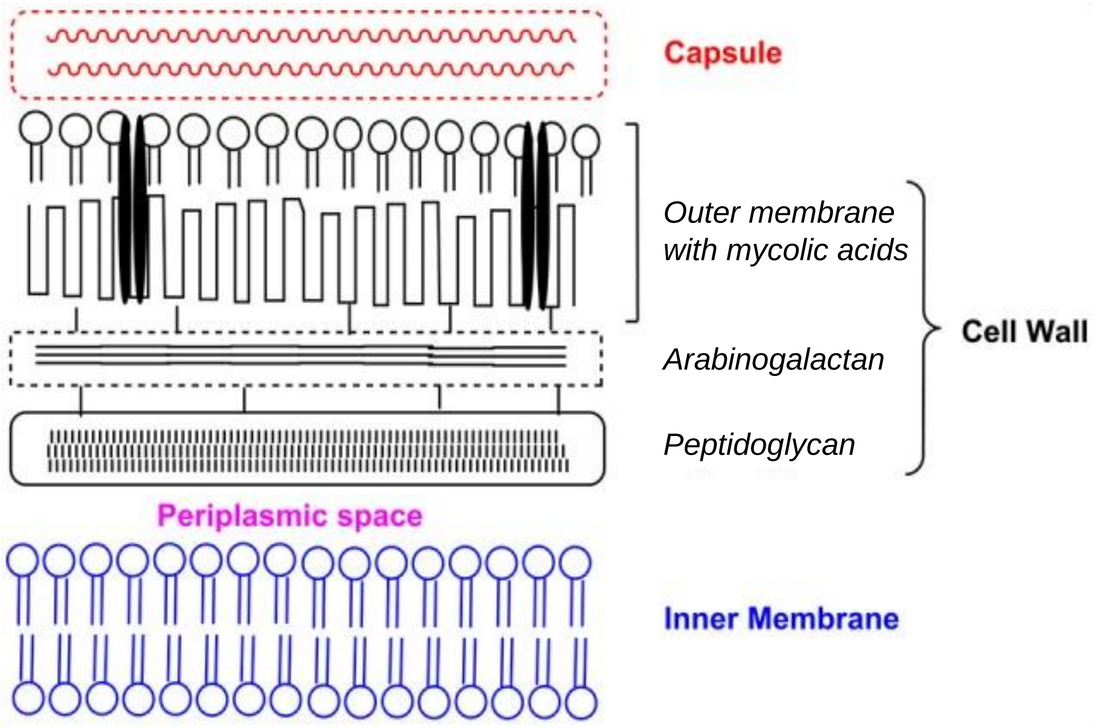


Figure 1.1: Schematic representation of a mycobacterial cell envelope. Adapted w/o permission from Chen *et al*. 2018 [14]

arabinomannans, mannans, and lipoproteins [15]. The inner membrane is a phospholipid bilayer, composed of glycerophospholipids and phosphatidyl dimannosides [16]. The cell wall of Mtb is a tripartite structure that consists of an outer membrane, containing long chain mycolic acids, that are covalently linked to the underlying arabinogalactan-peptidoglycan complex [17]. The abundance of lipids, and the presence of these long chain mycolic acids in the envelope of Mtb is considered responsible for the acid-fast staining property [18]. Acid fastness is the principle behind the Ziehl-Neelsen (ZN) mycobacterial staining technique, which has historically been used as a means of diagnosing clinical Mtb infection [19] (**Figure 1.2**). It is however noteworthy that ZN stain is more often retained by Mtb bacilli that are actively replicating, and dormant bacteria are less likely to stain as ZN positive [20, 21]. It is also noteworthy that deletion of *KasB*, the gene required for mycolic acid synthesis,

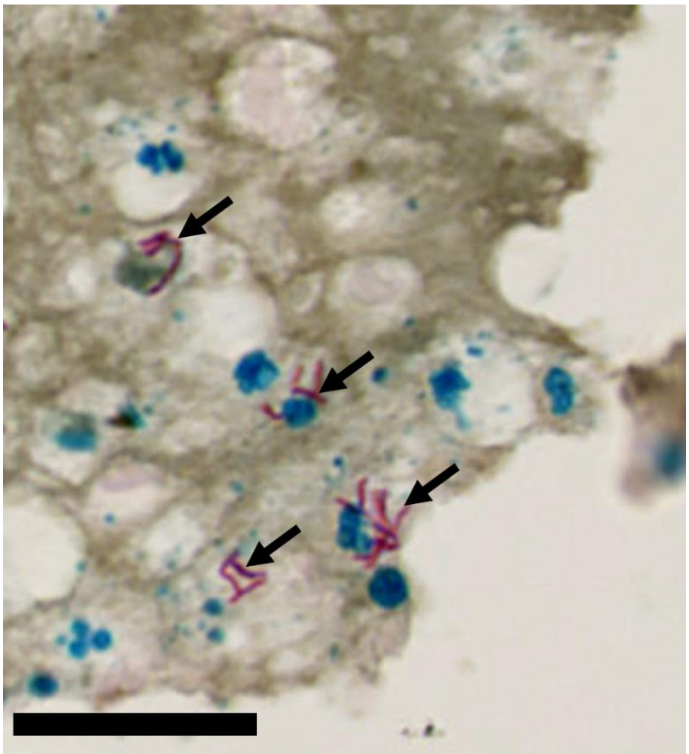


Figure 1.2: ZN stained Mtb bacilli in resected human lung tissue.

Mtb bacilli are stained magenta and indicated by black arrows. Cell nuclei are stained blue. Scale bar = 25 µm.

and thereby acid-fastness, both reduced active disease in mice infected with *kasB* deficient Mtb and abrogated bacterial cording, an aggregated mechanism of cellular growth, seen in replicating Mtb [18].

1.1.3 Mycobacterium tuberculosis aggregation

Mtb naturally aggregates during replication unless grown in the presence of a detergent [22, 23]. Robert Koch first described this behaviour when saying that Mtb bacteria "form small groups of cells which are pressed together and arranged in bundles" [4]. When grown in 7H9 culture media Mtb displays a mechanism of cell growth, termed cording, in which bacteria replicate and remain in close association with one another in filamentous "cords" (**Figure 1.3 A adapted from kaslum *et al.*** [24]). Cording has been shown in ex-vivo experiments using alveolar macrophages and has long been considered a virulence factor during Mtb infection [25-27] (**Figure 1.3 B adapted from Ufimtseva *et al.* [25]***)*.

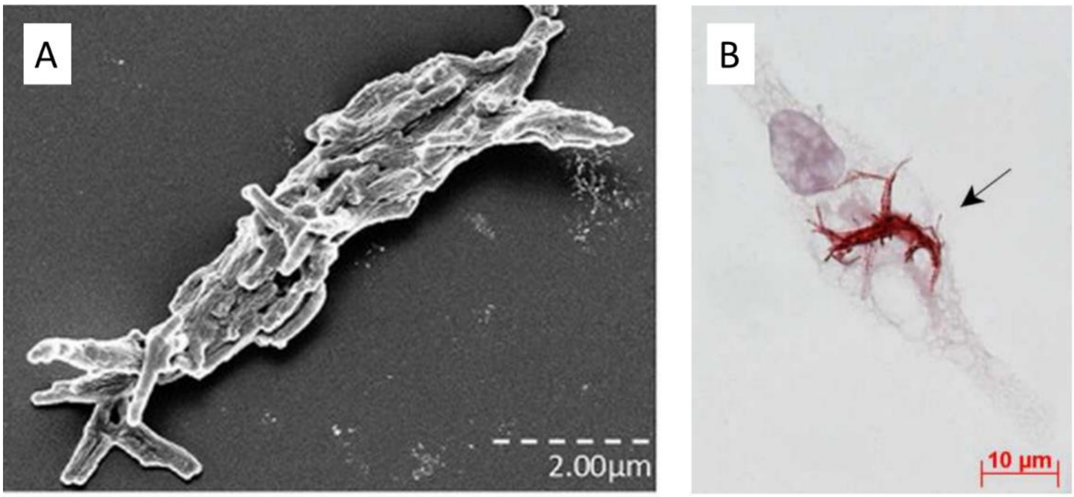


Figure 1.3: Cording aggregation in Mtb bacilli. Adapted w/o permission from Kalsum *et al.* 2017 and Ufimtseva *et al.* 2018 [24,25]

Scanning electron micrograph of aggregated cording Mtb bacteria (A) and ZN-stained cording aggregate in an alveolar macrophage (B).

Cording was recently linked to the progression to active TB disease in a mouse model that was capable of generating the complete spectrum of tuberculous lesions seen in human TB disease [28]. C3HeB/FeJ mice that were infected with a cording variant of H37Rv Mtb developed exudative lesions 17 days post infection. In contrast, mice that were infected with a non-cording variant of H37Rv had little to no detectable lesions [28]. This highlights the importance of Mtb aggregation in the context of an *in vivo* Mtb infectious lifecycle. It must be noted however, that bacterial aggregation may be the result of cording growth or general bacterial clumping mediated by hydrophobic interactions of the lipid rich bacterial cell envelopes [29]. Despite this distinction, Mahamed *et al*. demonstrated that Mtb bacterial aggregation, including clumping, remains an important factor in determining host cellular fate by showing that macrophages infected with Mtb aggregates undergo cell death more frequently than macrophages infected with multiple single Mtb bacilli [30]. What is less clear, is the stage(s) in the infectious lifecycle of Mtb at which aggregation exerts the most profound effect on the host/pathogen interaction. Aggregation is considered a virulence factor during Mtb bacterial growth, but the recent demonstration of Mtb aggregate transmission in bio-aerosols could indicate a role for Mtb aggregates during the transmission stage of the Mtb lifecycle as well [31].

1.1.4 The Mycobacterium tuberculosis infection life-cycle

Mtb is transmitted via airborne droplet nuclei during expectoration by Mtb infected individuals [32, 33]. The airborne Mtb bacilli are inhaled by new hosts and travel through the upper respiratory tract to implant on the surface of the human lung. At the lung surface, one of the first points of contact with the bacteria is the alveolar macrophage. These innate immune cells phagocytose the infecting bacteria and drive the recruitment of blood-derived macrophages, neutrophils and other immune cells to the site of infection [34]. This generates an inflammatory response that results in the development of a host protective structure, known as a granuloma, that attempts to "wall off" the pathogen (**Figure 1.4 adapted from Ramakrishnan *et al*. [35])** [36-39]. At first, this organised multicellular structure is an aggregation of innate responders that phagocytose and attempt to eradicate the pathogen. This early phase of the infection is typically characterised by bacterial replication and the killing of host phagocytes. Zebrafish models, using *Mycobacterium marinum* as the infectious agent, have demonstrated that the early granuloma environment may be conducive to pathogen replication by drawing in uninfected macrophages to the granuloma for productive infection by pathogen [40]. Inflammatory signals drive the recruitment of monocytes from the blood to the site of infection, which differentiate into specialized macrophage cellular subtypes. Such subtypes include foamy macrophages, epithelioid macrophages or multinucleated giant cells [41] (**Figure 1.4 adapted from Ramakrishnan [35]).** Dendritic cells are also recruited to the site of infection and, over time, an adaptive immune response develops. The development of an adaptive response can be delayed during an Mtb infection, but the recruitment of adaptive effector cell populations, such as T lymphocytes and B lymphocytes, during tuberculous disease progression usually correlates with enhanced pathogen control [42, 43]. These adaptive cells develop a cuff surrounding the granuloma. At core of the granuloma is a region that is typically characterized by a necrotic zone containing an abundance of dead host cells, cellular debris and Mtb bacteria which develops into a solid necrotic region known as caseum [35]. This caseus, hypoxic environment is considered to play an important role in transitioning the infecting bacteria to a dormant, non-replicating state that contains the pathogen but does not sterilize the infection [44]. Following granuloma development, an infection can remain stable and asymptomatic to the host, known as latent TB infection, or the infection may reactivate, escape immune containment and progress to active disease [45, 46]. Re-activation is typically characterized by liquefaction of the caseus, necrotic granuloma core, followed by rupture and cavitation [47]. This results in the expulsion of Mtb bacilli, through a productive cough, and spread of the pathogen to new hosts.

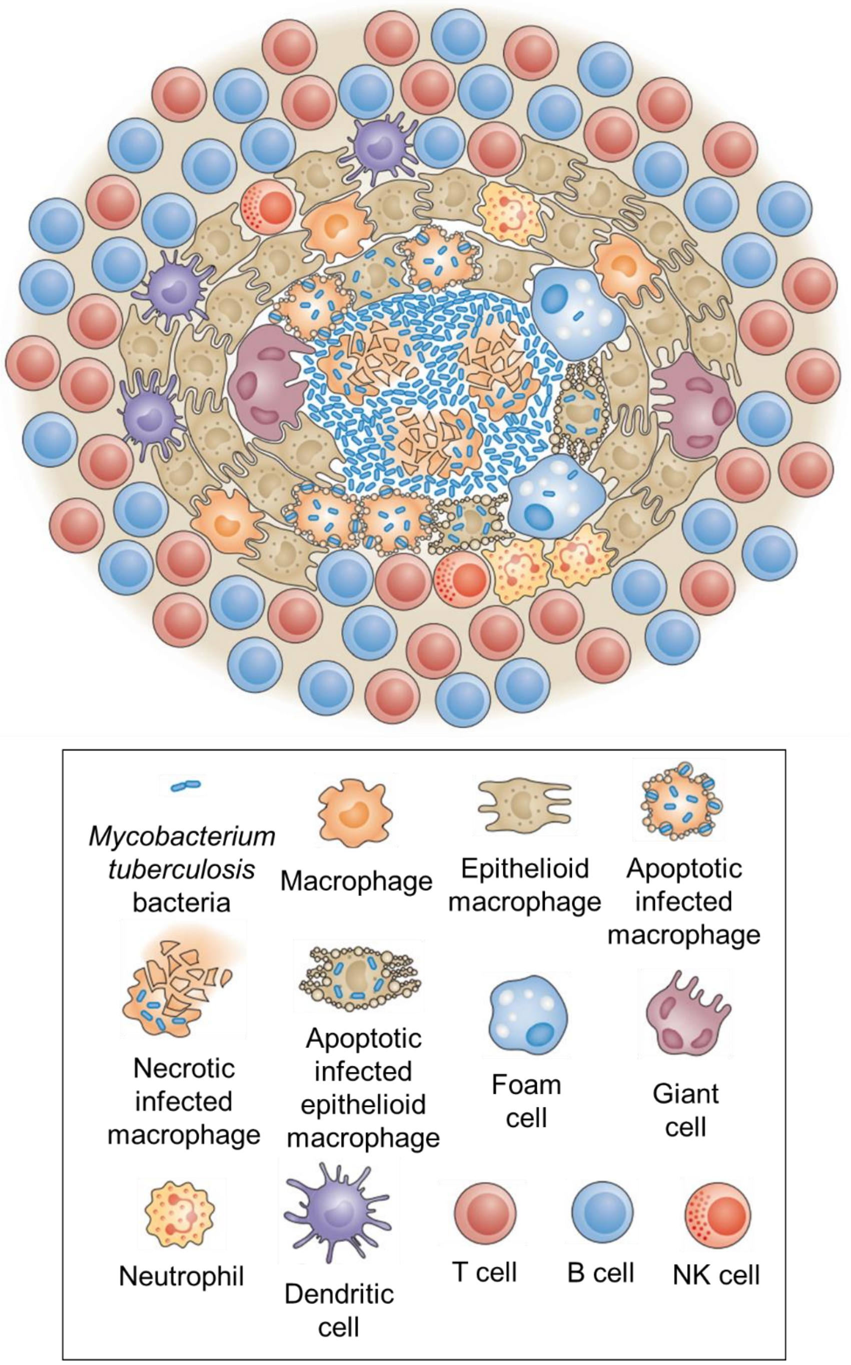

Mycobacterium tuberculosis bacteria
Macrophage
Epithelioid macrophage
Apoptotic infected macrophage
Necrotic infected macrophage
Apoptotic infected epithelioid macrophage
Foam cell
Giant cell
Neutrophil
Dendritic cell
T cell
B cell
NK cell

Figure 1.4: Schematic representation of the cellular composition of a classical tuberculous granuloma. Adapted w/o permission from Ramakrishnan 2012 [35]

A classical tuberculous granuloma is composed predominantly of epithelioid macrophages recruited to the site of infection. These tightly associated cells surround the infecting bacteria within a necrotic core abundant in cellular debris. Giant multinucleated cells and foam cells, rich in accumulated lipids, also develop from macrophages within the granuloma. Other innate effector cells, such as neutrophils and NK cells, are also drawn to the site of infection. Dendritic cell participation links the innate response to the adaptive response, resulting in a cuff of B and T cells organised around the granuloma structure.

1.1.5 Mtb infection outcomes are heterogenous

It is difficult to predict the outcome of a host immune response to Mtb. There is demonstrated variability between individuals in response to primary infection with the bacteria. Some individuals develop active TB disease after 1-3 years of infection, while others test positive for Mtb infection and never develop symptoms of TB [48]. Mtb infection can also transition to active disease after years of dormancy. This can occur as a result of a number of factors, such as immunocompromisation or treatment with TNF-α blockers [48, 49]. Furthermore, there is heterogeneity between granulomas within an individual lung. Some granuloma successfully contain the pathogen, while others within the same lung progress to cavitation [50]. While many known and unknown determinants likely affect the trajectory of an Mtb infection, Mahamed et. al demonstrated that Mtb aggregation state can affect individual cellular outcomes of infected macrophages. Infection with large Mtb aggregates elicited cell death that was similar to pyroptosis, a necrotic mode of cell death, in infected macrophages [30]. As one of the first cellular responders to an Mtb infection, a macrophage's ability to react adequately to a pathogen may be an important determinant for overall infection trajectory.

1.2 The Macrophage

1.2.1 Macrophage activation

Macrophages are a heterogenous and important population of innate immune cells that are among the first to respond to pathogens [51]. Tissue resident macrophages are a heterogenous population of cells, found in all mammalian organs, that function in clearance of cellular debris, immune surveillance, regulation of inflammation and other housekeeping functions [52]. Populations of alveolar macrophages, found in the lung, are first derived and established in the lung during embryonic development and are later self-replenished [53, 54]. Historically, this steady-state phenotype of tissue macrophages constitutes a metabolic state known as the M2 macrophage (alternatively activated macrophage) [55]. Conversely, the M1 differentiation state represents a macrophage metabolic state geared towards inflammation and anti-microbial activity (classically activated macrophage). This binary macrophage phenotype classification system has been considered an over-simplification, as macrophages grouped into the M2 phenotype have been found to encompass a wide range of biological functions, and macrophage activation has been proposed to exist on a more complex spectrum as a result of stimulation with a multitude of host mediated and environmental stimuli (**Figure 1.5 adapted from Mosser *et al*.**) [56, 57]. However, the M1 and M2 designations are useful when describing host macrophage responses to infectious stimuli. M2 macrophages can be driven towards the M1, classically

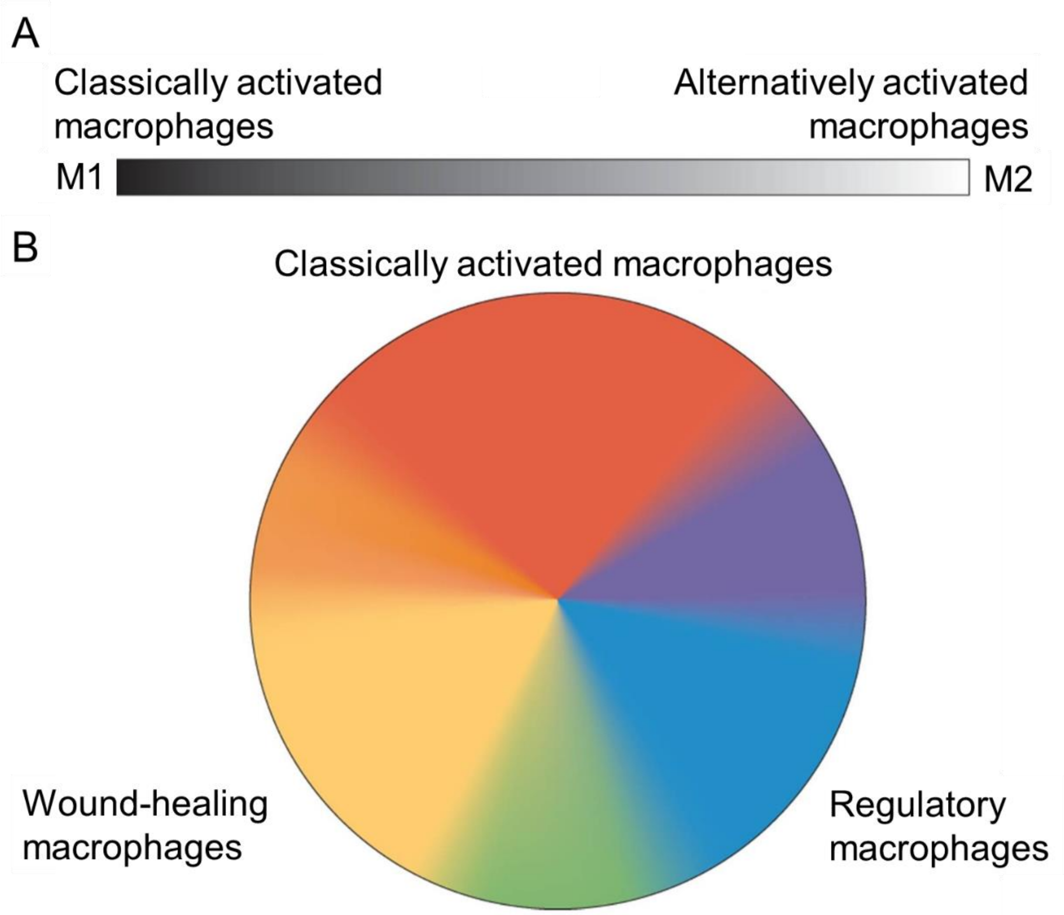

A
Classically activated macrophages
Alternatively activated macrophages
M1
M2
B
Classically activated macrophages
Wound-healing macrophages
Regulatory macrophages

Figure 1.5: The macrophage spectrum of activation. Adapted w/o permission from Mosser *et al*. 2008 [56]

Macrophages are defined as being classically (M1) or alternately activated (M2) (A) but have been proposed to develop more complex activation spectra, with associated biological functions, as a result of stimulation with a variety of environmental and host mediated stimuli (B).

activated, phenotype through exposure to IFN-γ or TNF-α [56]. Additionally, studies in mice have shown that alveolar macrophages, closer to the M1 phenotype, are more permissive to the growth of infecting Mtb than interstitial macrophages, a blood-derived M2 like macrophage, also found in the lung compartment [58]. Outside of tissue origin and activation state, macrophages may also differentiate into specific cellular subtypes during the course of an Mtb infection.

1.2.2 Macrophage subtypes during a granulomatous response

During mycobacterial contact at the surface of the lung, infected alveolar macrophages drive the recruitment of blood derived macrophages to the site of infection [59]. Here, macrophages often further differentiate into several subtypes during granuloma formation. This includes foamy macrophages, epithelial macrophages and giant cells [35]. Although the precise mechanism of differentiation to these cell types is poorly understood, some of their roles during Mtb infection have been identified. Foamy macrophages are characterized by excess lipid accumulation and are thought to represent an energy rich infectious niche for Mtb bacteria [60]. *In vitro* studies have also suggested that mycobacterial mycolic acids stimulate the differentiation of monocytes into foamy macrophages [61]. A study by Kim *et al*. demonstrated the spatial organisation of gene expression in a granuloma and showed that the central necrotic core, where Mtb bacteria are located, was characterized by an upregulation of genes involved in host lipid metabolism [62]. Epithelioid macrophages become tightly associated with one another and form a barrier around the infecting Mtb and are thought to be instrumental in granuloma formation [35]. Giant cells are the result of fusion between macrophages and have been shown to be deficient in the ability to phagocytose invading mycobacteria [63, 64]. The development of macrophage cellular subtypes during an Mtb infection may be an important determinant for infection trajectories, yet the individual macrophage is the most common point of first contact in the lung with the Mtb pathogen. Therefore, the ability of this cell type to detect and respond to a mycobacterial infection may be an important and early factor for determining infection outcomes.

1.3 Macrophage response to infection

1.3.1 Macrophage pathogen sensing

Macrophage function and activation state is influenced by the binding of substrate to cell receptors. Macrophages that perform homeostatic functions, such as removing cellular debris and phagocytosis of apoptotic host cells, use scavenger, complement, thrombospondin and phosphatidylserine receptors to detect ligands associated with these processes. These activities typically occur in the absence of immune cell signalling and do not generate an inflammatory response [65, 66]. Conversely, there are also ligands that elicit inflammation and an immune response when detected by macrophages [65]. These signals include bacterial surface proteins, bacterial or viral nucleic acids and are known as pathogen-associated molecular patterns (PAMPs). Other host-derived cellular stress signals, such as intracellular cytokines and debris generated as a result of traumatic cellular death, can likewise elicit a more inflammatory response when detected by macrophages and are known as damage-associated molecular patterns (DAMPs) [65]. Toll like receptors (TLRs) 1,2 and 4, found on the macrophage surface and within phagosomes, are PAMP receptors that function to detect Mtb, and other pathogens, when phagocytosed by patrolling macrophages [67-72]. Detection of pathogen by TLRs results in the induction of a protective inflammatory response that includes the production of cytokines such as TNF-α [69]. However, this protective signalling cascade, that involves the MYD88 adaptor molecule, has been shown to be dependent on phagosome acidification in macrophage infection models using *Staphylococcus aureus* [73]. Macrophage phagosome acidification/maturation is a process which Mtb is known to inhibit [74-76].

1.3.2 Phagolysosome acidification and maturation in response to pathogen

Mtb inhibits the acidification of phagolysosomes during infection [74-76]. Phagosomes containing internalized pathogen becomes acidified via incorporation of a proton pump, V-ATPase, acquired from lysosomes into the nascent vesicle [77]. The resulting lowered pH within the compartment has historically been considered an antimicrobial defence mechanism [78]. However it has been demonstrated that Mtb survives at pH levels found in acidified phagosomes [79]. Acidification of the phagosome has been shown to be synergistic with other antibacterial mechanisms and has been recognised as an important signalling event in response to bacterial pathogen [77]. A lower intraphagosomal pH facilitates the action of digestive enzymes found within the phagosome. These enzymes liberate bacterial ligand for binding to TLRs [73]. TLRs in turn initiate a signalling cascade that can initiate an inflammatory and antibacterial response [80]. The MyD88 adaptor molecule has been demonstrated to play a critical, acidification-dependent role in this signalling cascade [73]. Inhibition or interference with this process therefore represent an opportunity for Mtb to generate sub-optimal host responses.

1.3.3 Inhibition of phagosome maturation by Mtb

Mtb is thought to interfere with phagolysosome maturation via a number of mechanisms [81-87]. Mannose-capped lipoarabinomannan is a lipoglycan, found in association with the mycobacterial cell wall, that has been shown to prevent phagosome maturation via inhibition of the membrane trafficking lipid phosphatidylinositol 3-phosphate (PI3P). This prevents the acquisition of lysosomes by nascent Mtb-containing phagosomes [81-83]. Additionally, the bacterial secreted phosphatase SapM has been shown to slow phagosome maturation via cleavage of PI3P [84]. Furthermore, Mtb can directly interfere with V-ATPase assembly and thereby directly inhibit phagosome acidification using the bacterial secreted protein tyrosine phosphatase (PtpA) [85]. Another bacterial lipid, trehalose dimycolate, has been shown to slow phagosome maturation when coated onto magnetic beads and phagocytosed by macrophages [86]. The inhibition of acidification in the macrophage phagosome is also an important step in the ESX-1 mediated rupture of Mtb containing phagosomes [87]. It is noteworthy that phagosome maturation and acidification can be restored in IFN-γ activated macrophages and that resting macrophages are more susceptible to inhibition of the phagosome maturation process [88, 89]. However, Mahamed *et al*. showed that activated MDMs, that were infected with Mtb aggregates, became acidified prior to undergoing necrotic cell death [30]. This indicates that Mtb aggregates were able to overcome the macrophage response even after phagosomes become acidified.

1.3.4 The Macrophage transcriptional response to infection

The macrophage transcriptome exhibits profound remodelling as a result of cellular exposure to bacteria. Microarray studies have shown that through exposure to a broad range of bacterial components, including Mtb antigen, host macrophages upregulate a common set of genes that activate the cells and prime them for an immune response [90]. A study using Cap Analysis of Gene Expression (CAGE) showed that macrophage infection with Mtb resulted in a broad transcriptional response, upregulating genes for transcription factors in both M1 and M2 macrophage activation states [91]. Gene cassettes that were upregulated encompassed a wide array of biological functions such as pro-inflammation, cell death, negative regulation of apoptosis, and cytokine or chemokine signalling targeting other immune cells, such as neutrophils [90, 92]. A study involving M1 polarized macrophages demonstrated that activated macrophages infected with Mtb induced many inflammatory gene pathways, and that the effects of Mtb infection on macrophage transcriptional profiles mimicked those seen for activation by IFN-γ [93]. Ontogenetically distinct macrophage populations show differing metabolic transcriptional profiles during Mtb infection, with alveolar macrophage populations being more permissive to Mtb growth and favouring fatty acid oxidation rather than glycolysis as seen for interstitial macrophages that restricted bacterial replication in infected mice [58]. These transcriptional responses to infection can therefore be important determinants of antimicrobial activity and cellular infection outcomes.

1.3.5 Initiation of macrophage antimicrobial effector functions

Transcriptional initiation of antimicrobial effector functions can be elicited through a number of pathways (**Figure 1.6 adapted from MacMicking** [80]). IFN-γ has been identified as a primary inducer of macrophage activation, and is typically produced by other immune cells, such as CD4+ T-lymphocytes, in the context of a bacterial infection [94, 95]. Stimulation with this cytokine initiates transcription through a STAT1 mediated pathway (**Figure 1.6 adapted from MacMicking** [80]). TNF-α and IL1-β are also important cytokines that have been implicated in the control of mycobacterial infections.[96, 97] TNF-α induced responses result from binding of TNF-α to the type 1 TNF receptor (TNFR1) and can share a common signal transduction pathway with receptor binding of IL-1β and IL-1α by IL-1R1/IL-1RAcP (Interleukin receptors) and binding of Mtb derived ligands to TLRs 1,2 and 4 (**Figure 1.6 adapted from MacMicking** [80]). It is also noteworthy that the IL-1R1 and TLR receptor signal transduction passes through the MyD88 adapter molecule pathway, which has been shown to be dependent on phagosome acidification for effective pathogen processing, and that IL1-B can upregulate both TNF-α secretion and TNFR expression [73, 98]. Induction of these pathways leads to the production of antimicrobial effectors that include a variety of reactive oxygen species (ROS) or reactive nitrogen species (RNS), antimicrobial peptides (AMPs), or the generation of an autophagic response (also known as xenophagy), a protective cell death mechanism that targets invading bacteria [99-101]. Although binding of these receptors mediates important antimicrobial activities, these responses do not always successfully eradicate the invading pathogen. IL-1β has been shown to play an important role in the induction of an inflammatory mode of cell death, known as pyroptosis, that facilitates the rupture of an infected cell and subsequent spread of Mtb bacteria to neighbouring cells [102]. TNF-α, while important in antimicrobial signalling cascades, can have pleiotropic effects in the context of an Mtb infection (see below).

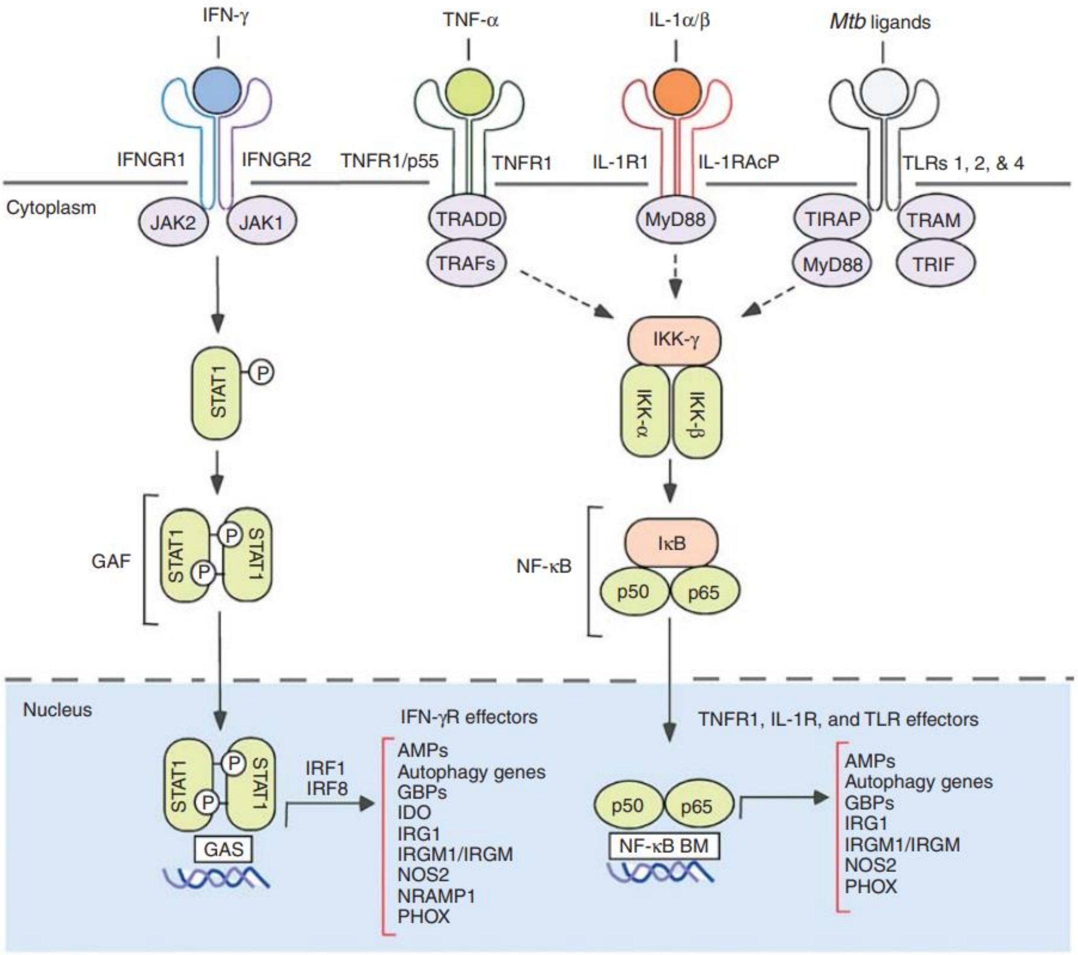


Figure 1.6: Schematic overview of signalling cascades involved in initiation of macrophage antimicrobial effector functions. Adapted w/o permission from MacMicking 2014. [80]

NF-κB transcriptionally mediated macrophage antimicrobial effector mechanisms can be elicited through stimulation of TLRs 1,2 and 4 by Mtb ligands, via stimulation of interleukin receptors (IL1-R1/IL-1RAcP) by IL-1β/α or by TNF-α binding to tumour necrosis factor receptors (TNFR1/p55). STAT1 mediated antimicrobial effector functions can be elicited via stimulation interferon gamma receptors (IFNGR1/IFNGR2).

1.3.6 TNF-α and the response to Mtb infection

TNF-α is responsible for generating a multitude of host responses to combat Mtb infection. TNF-α is predominantly produced by monocyte/macrophage cell types in response to cytokine or bacterial products but can also be produced by other immune cells such as T-lymphocytes, B lymphocytes and natural killer cells [103]. TNF-α is an inducer of the transcription factor NF-κB, which regulates the transcription of a large number of inflammatory gene products [104] (**Figure 1.6 adapted from MacMicking** [80]). TNF-α has also been demonstrated to be instrumental in the recruitment of leukocytes to the site of Mtb infection, as well as being implicated in the organisation of a granuloma into a structure capable of containing the pathogen [105, 106]. The importance of TNF-α is highlighted in the reactivation of tuberculous disease in patients receiving immunotherapy with TNF-α suppressors [107]. Excess TNF-α has also been implicated in programmed macrophage necrosis, an inflammatory mode of cell death that results in release of Mtb bacilli into surrounding tissue [108]. Interestingly, this cytokine has also been implicated in the induction of apoptosis, a mechanism of host cell death that can have anti-inflammatory and protective effects in the context of a microbial infection [109].

1.3.7 Macrophage death as a response to infection with Mtb

Induction of cell death can result in enhanced control of invading pathogens and some mechanisms of host cell death can be more beneficial than others. Macrophages infected with virulent Mtb undergo more necrotic cell death than macrophages infected with avirulent Mtb, where apoptosis was the predominant mechanism of cell death [110, 111]. These two modes of host cell death can be categorised as inflammatory and non-inflammatory mechanisms respectively. Inflammatory modes of cell death include pyroptosis, necroptosis or trauma induced necrosis, while non-inflammatory modes of cell death include mechanisms such as apoptosis or autophagy [65]. Non-inflammatory mechanisms of cell death afford other host phagocytes an opportunity to contain Mtb infection without excess inflammatory immune activation. This is mediated via retention of host cellular membrane integrity for a period after cell death induction. This prevents the release of DAMPs into the extracellular environment and has been suggested to have a protective effect in the context of an Mtb infection [65, 112, 113] (**Figure 1.7 adapted from Kono and Rock**). A recent study also showed that Mtb elicits a broad range of both inflammatory and non-inflammatory host cell death responses, including apoptosis, while simultaneously downregulating apoptosis, resulting in a net enhancement of bacterial survival [114]. Interestingly, apoptosis has also been proposed as a mechanism by which Mtb can promote bacterial infection of additional host phagocytes when infected with a high MOI of Mtb [115]. The release of DAMPs, cytokines and chemokines, following

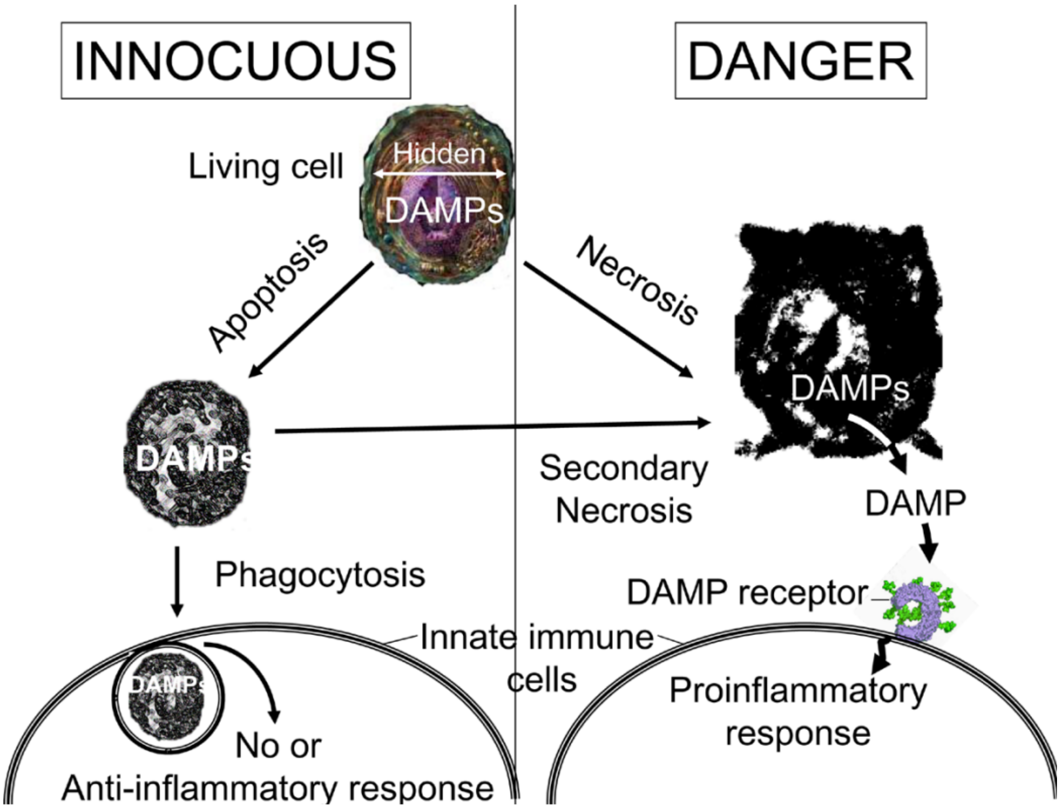


Figure 1.7: Schematic of consequences of pro and anti-inflammatory modes of cell death. Adapted w/o permission from Kono and Rock 2008. [65]

Apoptotic cell death results in retention of membrane integrity by dead host cells, allowing anti-inflammatory and innocuous phagocytosis by patrolling phagocytes. Necrotic mechanisms of cell death result in rapid cell membrane rupture and release of DAMPs into the extracellular space which can generate an inflammatory response in phagocytes detecting DAMPs.

inflammatory modes of cell death, into the surrounding tissue elicits the migration of additional phagocytes/immune cells to the site of infection and has been proposed as a mechanism through which Mtb draws new targets for infection and spread of bacteria. [40, 65, 116]. A phenomenon termed serial killing, shown by Mahamed *et. al*, has demonstrated that a single Mtb aggregate can kill several infected phagocytes in succession with increasing probability of eliciting inflammatory host cell death [30].

1.3.8 The Macrophage response to infection with Mtb aggregates

A study by Mahamed *et al*. investigated the effects of Mtb MOI and aggregation state on macrophage death when infected with Mtb bacteria. Blood-derived human macrophages showed increased cell death when infected with aggregated Mtb. This, and other, studies showed that a higher Mtb MOI elicits host cell death faster and more frequently in macrophages [30, 114, 117] (**Figure 1.8, adapted from Mahamed *et. al* [30]**). Interestingly, MDMs that were infected with a single large aggregate of Mtb had a higher probability of undergoing cell death than MDMs that phagocytosed a similar number of Mtb bacteria in the form of several smaller aggregates or multiple single bacteria [30] (**Figure 1.9, adapted from Mahamed *et. al* [30]**). The study also made several other observations relating to the infection dynamics of Mtb aggregates. Mtb aggregates that infect and kill host macrophages grow preferentially within the remains of dead MDMs, relative to extracellular media or within live MDMs [30]. A single Mtb aggregate has the potential to kill more than one host cell and can elicit death sequentially in MDMs that phagocytose the same aggregate in a process described as “serial killing” [30] (**Figure 1.10, adapted from Mahamed *et. al* [30]**). Once a Mtb aggregate has infected and killed a MDM (and proliferated in the cellular remains), it is equally or more cytotoxic to other phagocytes that pick up the same bacterial clump [30]. Finally, the macrophage death elicited by high MOI Mtb infection more closely resembles necrotic mechanisms of cellular death, such as pyroptosis, and not apoptosis [30]. This shows that the cellular fate of a macrophage infected with

Mtb can be affected by physical parameters of the infecting bacteria such as bacterial number and aggregation state. Elucidating the early transcriptional response of macrophages that are infected by Mtb aggregates could therefore reveal signalling events that characterise a failed cellular response.

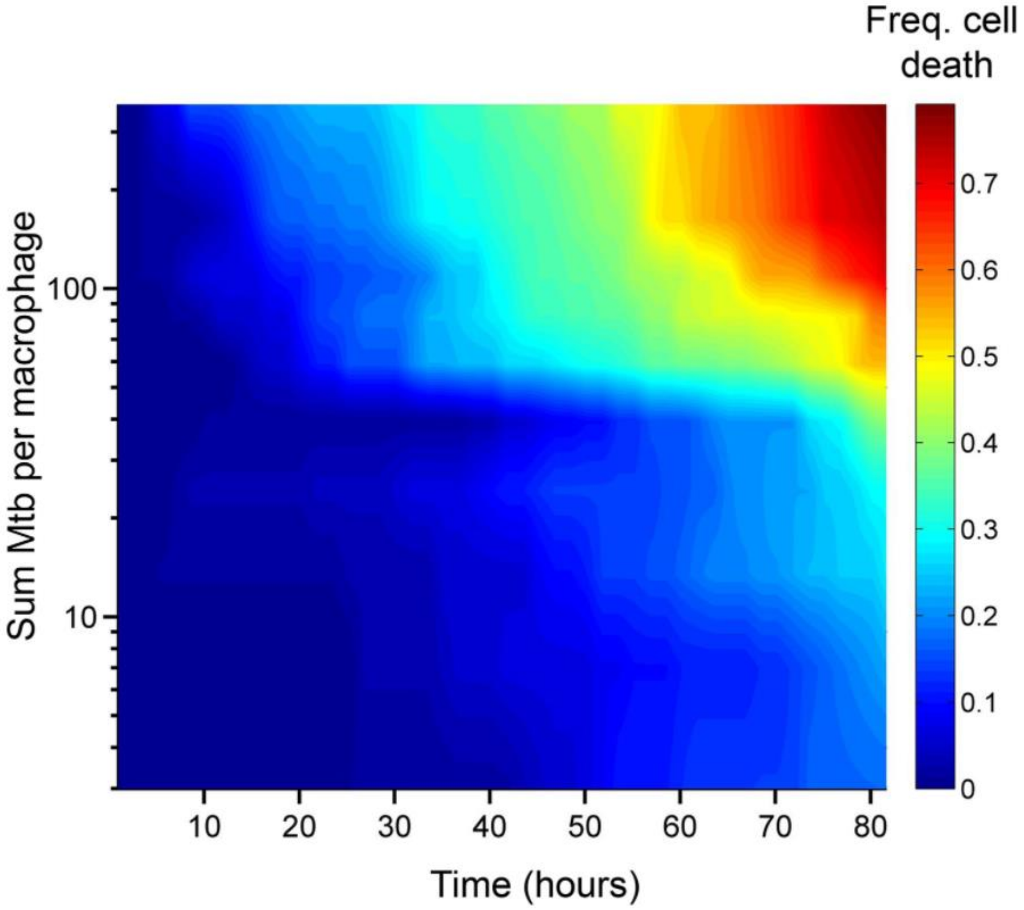


Figure 1.8: The number of infecting Mtb bacilli determines probability of phagocyte death. Adapted w/o permission from Mahamed *et al.* 2017. [30]

Frequency of cell death (colour scale bar) in Mtb infected MDMs (n = 720) increases with increasing number of phagocytosed Mtb bacilli. There is a marked decrease in the time (x - axis) to cell death and an increase in frequency of cell death, at ~30 infecting bacilli, relative to macrophages infected with a lower number of Mtb bacilli (y-axis).

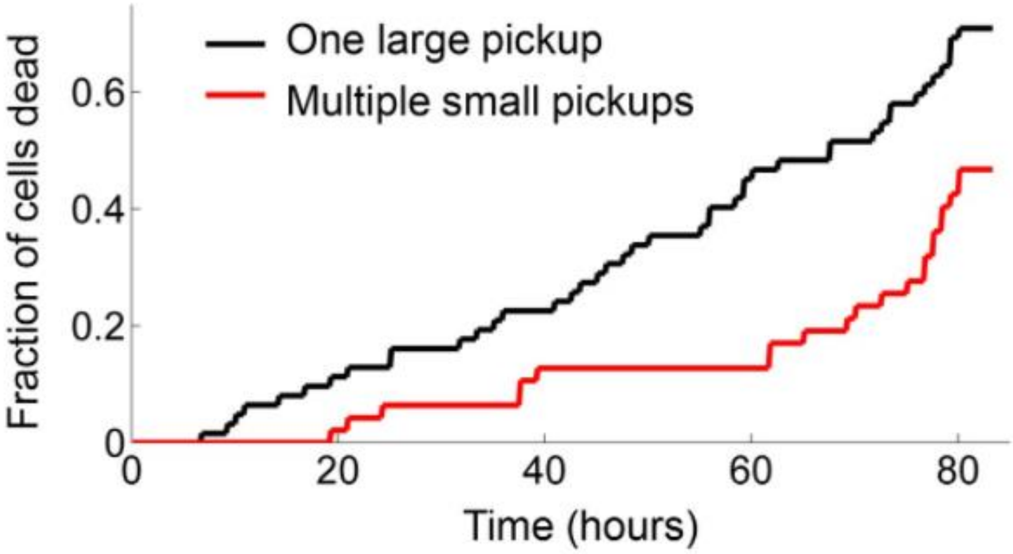


Figure 1.9: Single large aggregates of Mtb are more cytotoxic than multiple single Mtb. Adapted w/o permission from Mahamed *et al.* 2017. [30]

Total fraction of cell death (y-axis) for macrophages internalizing a single large aggregate (black line, n = 62) was 71%. Total fraction of dead macrophages internalizing a similar number of Mtb bacilli as multiple smaller aggregates or singles (red line, n = 47) was 47% in comparison.

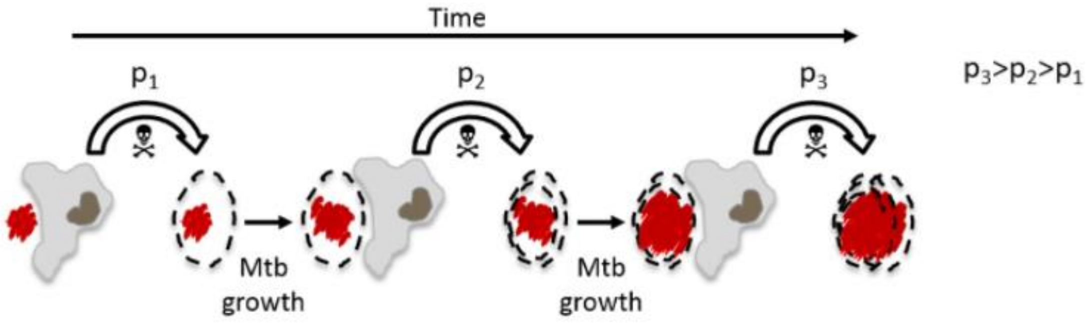


Figure 1.10: Mtb aggregates kill multiple phagocytes in successive phagocytic events. Adapted w/o permission from Mahamed *et al.* 2017. [30]

Infecting Mtb aggregates (red) infect and kill macrophages with probability P_1 and increase in size/bacillary number within the dead macrophage. This infection cycle repeats for subsequent phagocytes internalizing the bacterial clump, with probability of death from infection $P_n > P_3 > P_2 > P_1$.

1.4 RNA sequencing and expression analysis

RNA expression analyses have been used to great effect in understanding cellular transcriptional responses to infection. Microarray data has historically provided insight into expression patterns in a targeted fashion, where the expression of specific genes can be investigated [92, 118]. The advent of high throughput sequencing technology, such as that applied on platforms like the Illumina NextSeq, provides a hypothesis free means to investigate cellular expression patterns without limiting the scope of the genes being analysed [119]. Additionally, the variety of software packages that can be employed to investigate the resultant read-count data has increased in recent years. The R programming platform supports a number of bioinformatics packages that are specifically developed for processing cellular population expression data. Such packages include DESeq2, EdgeR, limma and others. DESeq2 has demonstrated good performance, in terms of differential expression (DE) analysis, relative to other packages and comes with a variety of additional tools for transcriptional data analysis [120, 121]. RLog normalization is a robust algorithm used by DESeq2 that minimizes the disproportionate effects of genes with either very high or very low read counts by coercing the data to have a similar dynamic range (homoscedastic). This regularizes the data so that it is better suited to interpretation using dimensionality reduction techniques, such as the principal component analysis (PCA), employed during expression clustering analysis [122]. During DE analysis, which directly compares expression patterns between treatment conditions, DESeq2 makes statistical inferences using a negative binomial distribution model. It is noteworthy that the DE analysis, as applied by DESeq2, constitutes a separate analysis technique, that uses a log fold change shrinkage technique to reduce the influence of genes with low statistical significance, that is distinct from the RLog normalization applied during expression clustering [122]. Sequencing technologies are an instrumental arm of bioinformatics aimed at investigating cellular genetic phenomena. The analysis of biological image databases represents another facet of bioinformatics aimed at extracting information resulting from data generated from imaging techniques such as microscopy.

1.5 Image analysis in infection biology

1.5.1 Microscopic image analysis

Much of our understanding of Mtb comes from the study of stained tissue slides [123]. Microscopic analysis is one of the earliest tools used to investigate microbial pathogens and forms the basis of some of our fundamental observations on the morphology of Mtb during infection as well as the host granulomatous response to infection. The advent of digital imaging platforms, tools and experimental models to directly image interactions between host and pathogen have also provided deeper insight into TB disease. Transparent Zebrafish models, in conjunction with fluorescent time-lapse microscopy have shown real-time granuloma development in response to infection with *Mycobacterium marinum* [40, 42]. The 3D structure of a granuloma in an Mtb positive human lung, in a departure from the classical spherical granuloma model, has been revealed to be more like that of a branching “ginger root” structure using digital reconstruction of CT scan images [124]. Mahamed *et al*. used a confocal fluorescent time-lapse microscopy system to investigate the macrophage death response following infection with Mtb aggregates [30]. Software platforms, like QuPath or ImageJ, afford the user a wide variety of tools with which to examine pathological features in histological tissue sections [125, 126]. Most approaches used for quantifying bacteria during microscopic analysis have been applied to *in vitro* bacterial culture or sputum smears, as opposed the more complex and visually heterogenous environment of a histological tissue slide [127, 128]. The size and complexity of these images, and the effort required for manual quantification of any stained bacteria that might be found in them, make these datasets good targets for the incorporation of machine learning (ML) techniques to automate image interrogation as far as possible.

1.5.2 Machine learning and digital image analysis

Deep learning machine models have been elemental in medical image analysis [129-132]. ML has been applied to tasks such as image segmentation, anatomical structure identification, diagnosis and classification in the context of histopathological staining [129]. For such analyses, a typical pipeline requires data pre-processing that involves sampling and feature extraction to obtain useful information from the raw image data [130]. Several types of ML models exist, such as deep convolutional neural networks (NN), support vector machines or polynomial regression.

1.6 Machine learning

Machine learning is a branch of artificial intelligence that employs a variety of mathematical models that use existing data to make predictions or classifications on unseen data (**Figure 1.11 adapted from Alzubaidi *et al***) [132]. Over time the model can improve in performance by incorporating more training data to improve the model fit. This is done by altering model parameters. For most ML models, this process can be broken into 3 steps during training, with each step represented by specific mathematical expressions/algorithms (**Figure 1.12**). This generally includes a hypothesis function, model cost function and a gradient descent algorithm. Once a model's parameters are trained, predictions/classifications are made in a single step (**Figure 1.12**).

1.6.1 Machine learning definitions

A data point, or observation, is a discrete unit of information. Observations can be composed of one or more metrics, known as features or variables, that describe its properties [133]. Features can include information such as length, width, or colour, of an observation depending on the dataset. Variables are denoted with an x in mathematical expressions seeking to model a dataset and are accompanied by coefficient terms. Theses coefficients are known as parameters and form the basis of a model's ability to change and fit to a dataset [133]. Parameters are denoted by an uppercase theta (θ) in model expressions. A hypothesis, denoted as $h_\theta(x)$, is the part of the model equation that describes the output or prediction of a model, given a set of input features and parameters in the expression. A hypothesis function is the expression that describes a model's prediction, given a set of input features and parameters [133]. **Equation [1]** describes a linear regression model, where the $h_\theta(x)$ (hypothesis of x) is equal to θ_0 plus the product of $\theta_1 \cdot x_1$.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 \qquad [1]$$

ML models change the value of parameters in the hypothesis function to enable a better fit of the training data. A model's performance, when compared to a dataset can be monitored using a cost function.

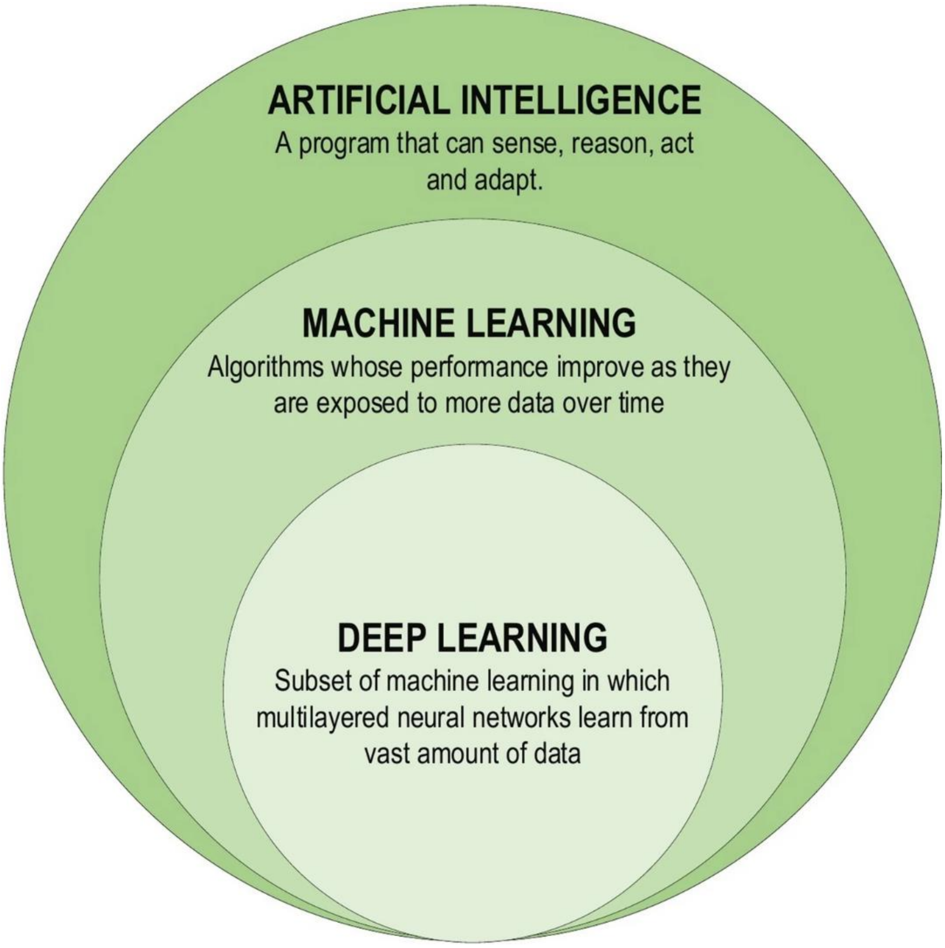

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act and adapt.
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amount of data

Figure 1.11: Machine learning is a branch of artificial intelligence. Adapted w/o permission from Alzubaidi *et al.* 2021. [132]

Artificial intelligence broadly encompasses mathematical models that dynamically react to data. Machine learning encompasses a group of algorithms whose performance can increase when exposed to increasing amounts of data. Deep learning describes a smaller group of complex neural network models that are typically trained using large datasets.

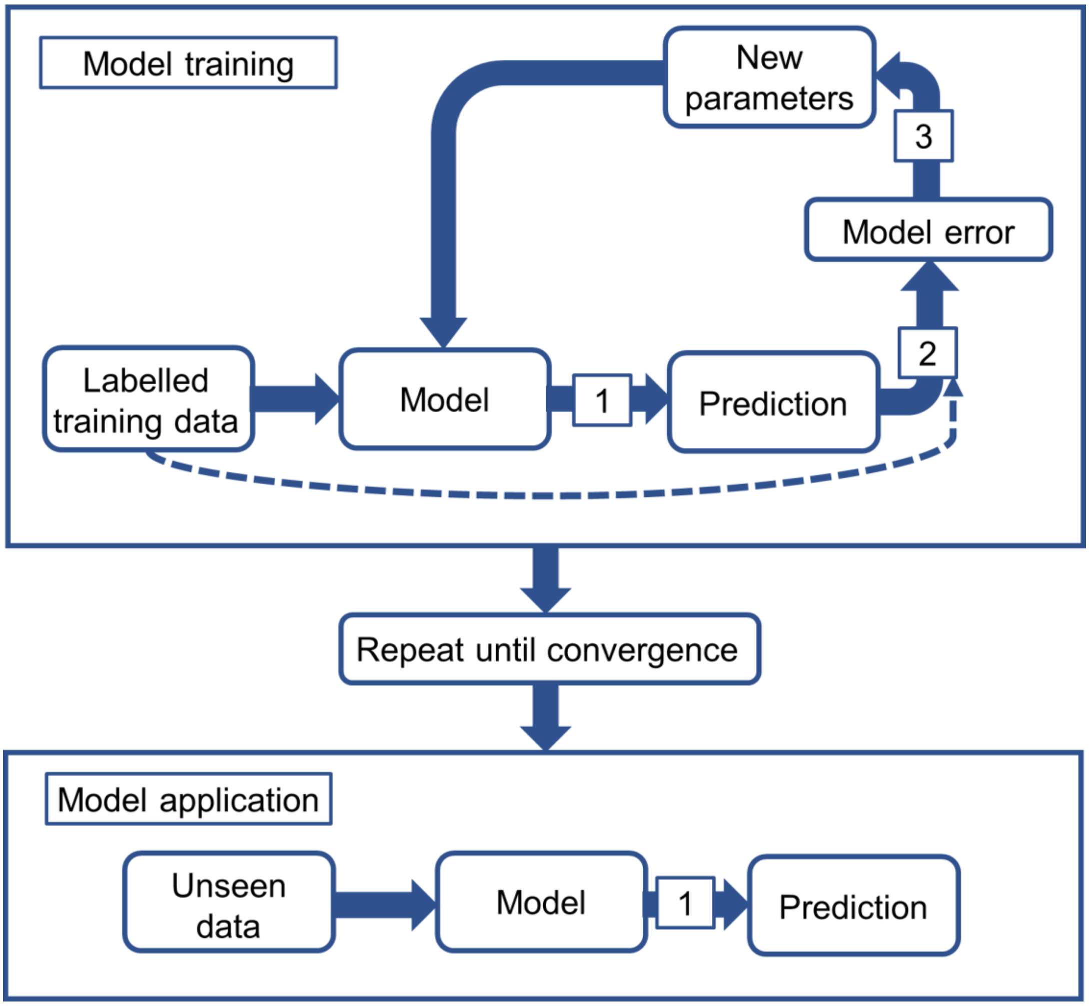


Figure 1.12: Schematic overview of machine learning model training and application

Labelled training data is used as input during model training. A model generates a prediction with a 1) hypothesis function. These predictions are then compared to data labels in the 2) cost function to estimate model error. New parameters are calculated from model cost using the 3) gradient descent algorithm and applied to the model. This cycle is repeated until the model achieves minimal cost (convergence). A trained machine learning model can then be applied to unseen data to generate predictions/classifications.

1.6.2 The model cost function

ML models are trained by incrementally reducing the cost of a model fit on a training dataset. Cost is calculated by comparing model predictions to pre-labelled data (also known as ground truths) and calculating an error term [134]. This process is analogous to a least squares error analysis. However, each type of ML model (including linear, polynomial or logistic regression, among others) has a specific cost function [134]. Varying model parameters will change the cost of a model, and there are parameter values at which model cost will be at its lowest. Finding the parameter values at which a model achieves the lowest possible cost represents the optimal possible fit of a model to a training dataset (**Figure 1.13**) [134]. Reaching this minimum cost value is the goal of ML models and achieving this known as model convergence. Several algorithms have been developed to automate this process, however, gradient descent is one of the most commonly employed.

1.6.3 Gradient descent

Gradient descent changes model parameters using an iterative, stepwise process. The gradient descent algorithm calculates the partial derivative of the cost function with respect to a particular set of parameter values [134]. This generates a tangent to the cost function from which a gradient can be extracted. This gradient provides directionality at each step and dictates whether the algorithm will add or subtract from the parameter values in the subsequent iteration of gradient descent (**Figure 1.13**) [134, 135]. The size of

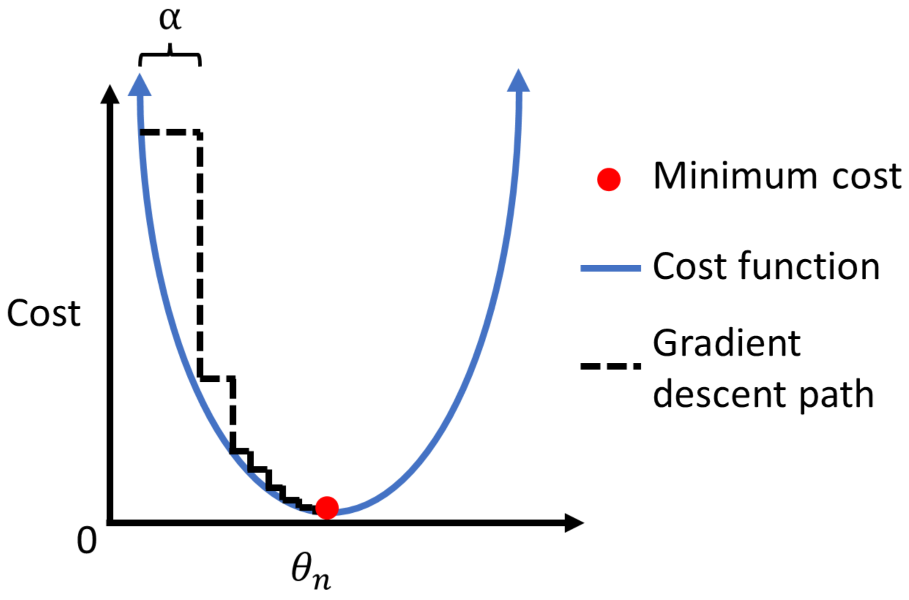


Figure 1.13: Model cost as a function of parameter value.

A ML model cost (blue line) will vary as a result of changing parameter values (θ_n). Gradient descent descent (perforated black line) varies model parameters to enable model convergence on minimal model cost (red circle).

each step of gradient descent is determined at each iteration, is dependent on model cost and is affected by the variable α (**Figure 1.13 α**). This means the pathway of gradient descent changes dynamically in response to changing model performance (**Figure 1.13 black perforated line**). The step size becomes smaller as the algorithm approaches a minimum cost value because the gradient is less steep, and the model cost is lower [135]. This makes the algorithm less likely to overshoot convergence but does not obviate the need to calibrate an appropriate value for α (**Figure 1.13 red circle**). α is an example of a hyperparameter. Hyperparameters are distinct from the parameters present in the hypothesis function of a ML model and are used to optimize and control the fit of a model to the training data.

1.6.4 Hyperparameters

1.6.4.1 Learning rate

The hyperparameter α is known as the learning rate [134]. If α is too large or too small a model may fail to converge on a minimum cost value. A large value for α can cause gradient descent steps sizes that are too large and generate an erratic gradient descent path. This can result in a failure to converge and potential underfitting of the data (high model variance), making it a poor predictor. Conversely, if α is too small the step size may be insignificant and the model may take too long to converge, likewise resulting in a poor fit to the data [135]. Notably, a perfect model convergence/fit may also be detrimental to performance when applied to unseen data. As this can lead to overfitting of the training data and poor model generalization when making predictions on unseen data (high model bias). This can be avoided by adding a regularization term to the model and controlled with the λ hyperparameter.

1.6.4.2 Model regularization

Overfitting can lead to poor model performance but can be controlled by model regularization [134, 136]. Biased ML models follow training data too closely (**Figure 1.14 Overfit Model**), while high variance models do not fit the data closely enough (**Figure 1.14 Underfit Model)**. By ensuring a model achieves minimum cost during gradient descent and then applying regularization term to “relax” the fit, a ML model can generalize better when the applied on real world data (**Figure 1.14 Regularized Model)**. A regularization term penalises model parameter values to limit the minimum cost a ML model can achieve during gradient descent. The hyperparameter λ is used to determine the magnitude of correction the regularization term applies to the model. This ensures that a ML model can only fit training data imperfectly, but still allows a model to fit the overall trend of the data and achieve better generalization when applied to unseen datasets.

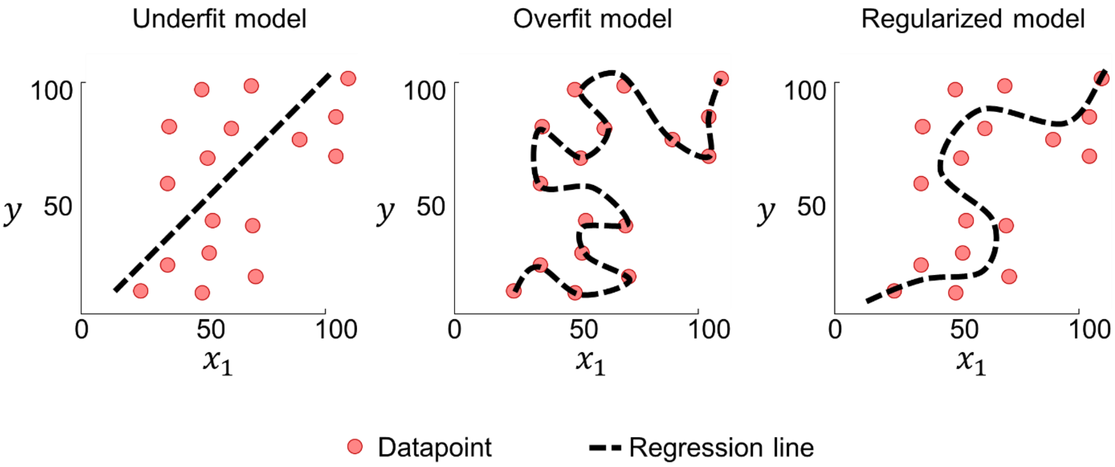


Figure 1.14: Overview of model bias and variance

ML models (black perforated line) that do not adequately model data (red circles) are Underfit. ML models that follow data too closely and do not perform well on unseen data. Regularized ML models fit the data imperfectly, but closely enough to enable good performance on unseen data.

1.6.5 Supervised and unsupervised learning

ML model training can be supervised or unsupervised [137, 138]. A supervised ML model takes advantage of pre-labelled data. In the case of a classification model, this means that training data has been annotated with the correct labels, typically through human data curation [139]. A model uses these labels as a “gold standard” to construct a decision boundary between labelled observations in the training data. This learned boundary can then be applied to unseen data to generate classifications. This method leverages human capacity to discriminate between data labels (via manual data curation) to generate better decision boundaries (**Figure 1.15 – Logistic regression**). On the other hand, an unsupervised learning algorithm uses training data that has not been manually curated [140]. The training data for unsupervised ML is labelled using clustering algorithms and takes advantage of natural differences in the data that cause it to group. This is typically accomplished using dimensionality reduction algorithms, such as principal component analysis (PCA) [140]. The current study makes use of supervised ML.

1.6.6 Prediction and classification

ML models can be used to predict values, based on the relationships between variables in a dataset, or they can be used to classify individual observations in a dataset into pre-defined groups [134]. The latter relies on two fundamental techniques; linear and logistic regression. Linear regression is used to generate predictions on unseen data by fitting a curve to a training dataset (**Figure 1.15 Linear regression**). Unseen observations are assigned a value based on their features in relation to a fitted regression line [134]. The hypothesis function in such cases returns a value in the same units as the response variable in the data to which the regression line was fit (**Figure 1.15 Linear regression hypothesis function**). Logistic regression is used to classify/assign labels to observations in a dataset (**Figure 1.15 Logistic regression)**. Observations are assigned a label based on their position relative to a decision boundary, a line that transects the data and described in the hypothesis function [134, 141]. (**Figure 1.15 Logistic regression decision boundary)**. The expression for the decision boundary is found in the exponential term of the base e (**Figure 1.15 Logistic regression, hypothesis function**). This expression also contains the terms for the dataset features and parameters **(Figure 1.15 Logistic regression, perforated red line**). The full hypothesis function for logistic regression takes the form of the sigmoid function, which asymptotes at 0 and 1 [141]. This means that the function evaluates to a probability that an observation is part of a particular label class, and therefore class labels can be assigned by user-defined thresholds. The logistic regression classification model is the basic unit for the individual nodes of a computational neuron, the key component of a convolutional neural network (CNN) classification model [142].

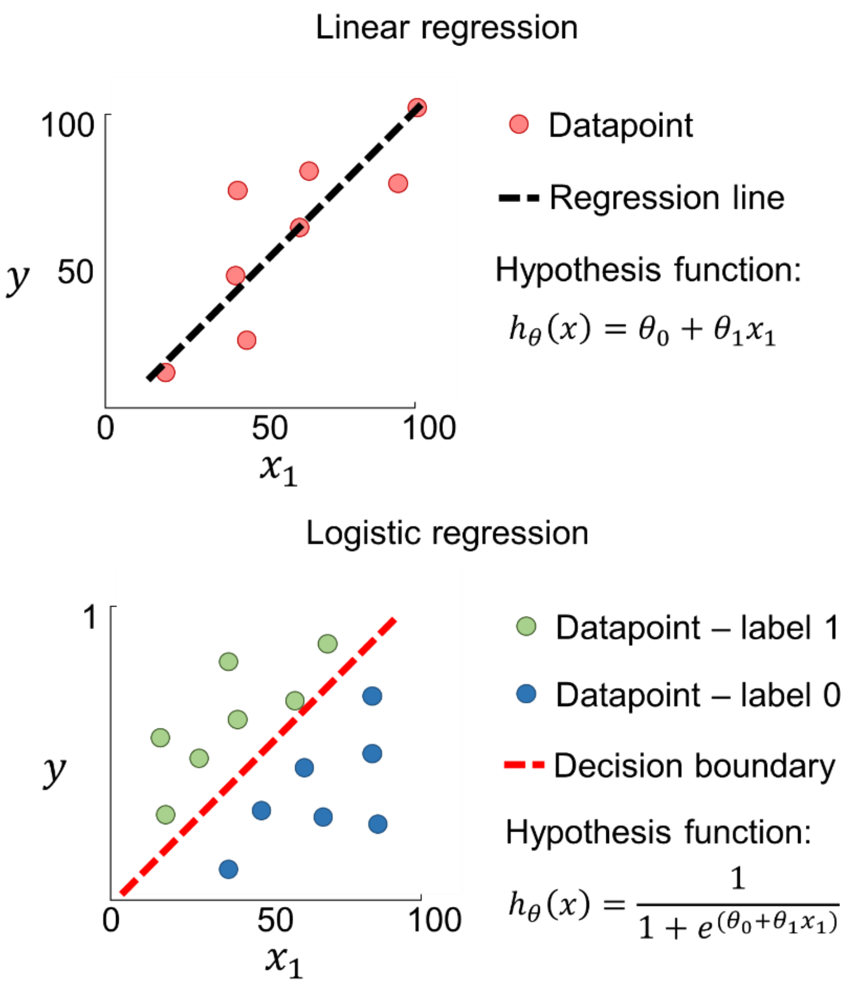


Figure 1.15: Comparison of linear and logistic regression

Linear regression (black perforated line) is used to generate predictions based on relationships between variables in the training data (red circle). Logistic regression is used to generate decision boundaries (red perforated line) that classify the data into different groups (blue circles and green circles).

1.7 Deep Learning

Deep learning is a branch of ML that deals with multi-layered neural networks that learn using large training datasets (**Figure 1.11**).

1.7.1 The computational neuron

CNNs are composed of basic computational units known as neurons. Neurons can have varying composition, but all have the same fundamental structure that is composed of an input layer, an output layer and one or many hidden layers (**Figure 1.16**) [143]. The structure of a computational neuron can be compared to a biological neuron. The input layer of a computational neuron is where data is input into the CNN algorithm structure (**Figure 1.16 - panel 1**). This is analogous to the dendrites of a biological neuron receiving signals from other cells. In the case of the computational neuron, the signals received in the input layer are the features of an observation in a dataset (**Figure 1.16 panel 1, x_1,x_2 and x_3**). These features are then multiplied by parameter matrices and used as input for logistic regression functions. The output of these functions constitutes what is known as the activations in the hidden layer of the neural network (**Figure 1.16, panel 2, a_1^1, a_2^1 and a_3^1**) [138].

Individual activations are commonly referred to as nodes. Each node uses all features from an observation in a dataset, multiplied by a unique parameter matrix, to calculate an activation value. Therefore, the output of each node is a unique value that incorporates all the data available for a single observation. This allows a computational neuron to form a unique hypothesis at each node of a network [143]. These nodes in turn form a matrix which is multiplied by another unique parameter matrix and used as input for a final, single logistic regression node (**Figure 1.16 panel 3**). The output of this final node is the hypothesis/classification for the entire computational neuron. For a CNN the calculation that evaluates to the overall classification/hypothesis for a neuron, beginning at the input layer and ending at the output layer, is known as forward propagation [134, 144]. The cost calculation and gradient descent algorithms for CNNs are also unique. The calculation of model gradients, used during gradient descent, for CNNs is referred to as backpropagation [145, 146]. A number of variations on the CNN model exist and can be applied in data-specific contexts. Recurrent Neural networks, for example, can take a variable number of features as input [132]. For the purposed of the current study however, where there is a defined set of features extracted from an image, a standard CNN was most suited.

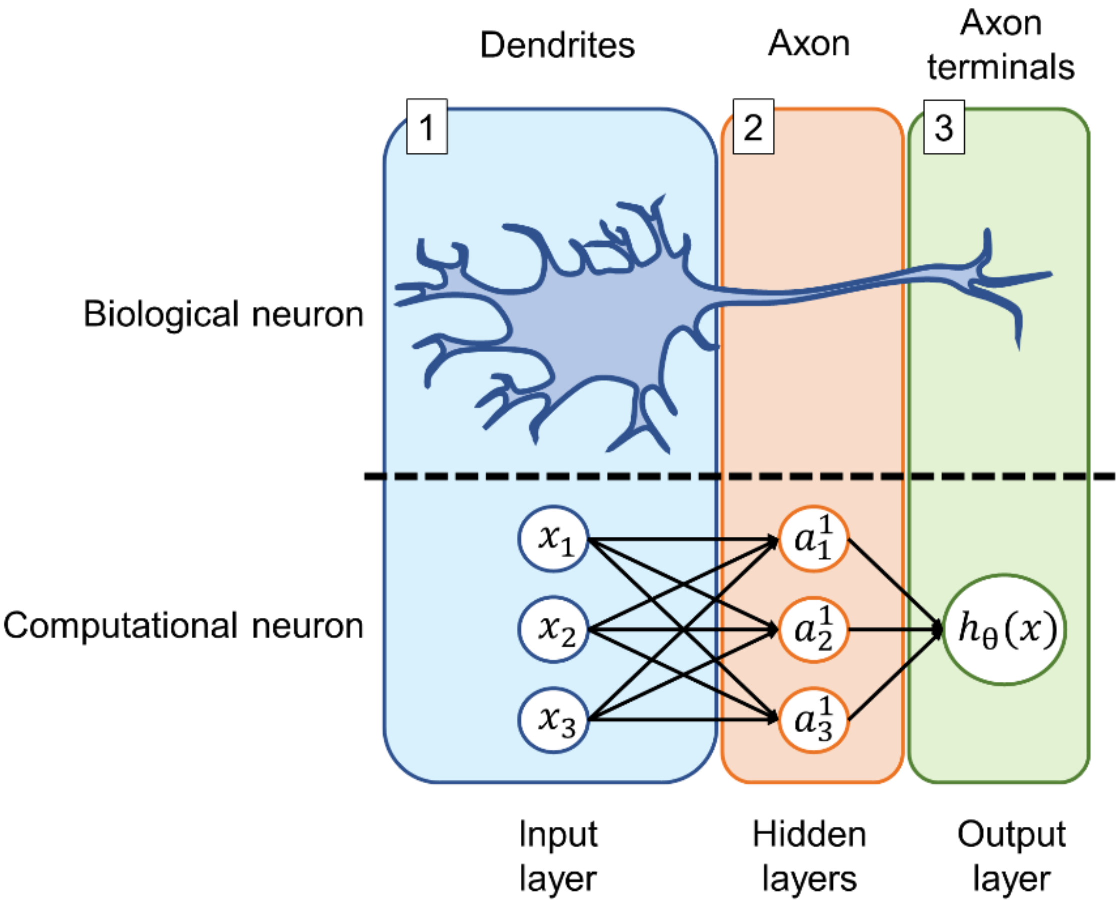


Figure 1.16: Schematic overview and comparison of computational and biological neurons

A computational neuron accepts data as input in the input layer, where a biological neuron receives signals at the dendrites (1). The hidden layers of a computational neuron generate unique hypotheses in the hidden layers where biological neurons pass signal along an axon (2) The output layer of a computational neuron generates a classification for data received in the input layer, where a biological neuron transmits signals to other neurons at the axon terminals (3).

1.8 Aims

Macrophages are an important population of innate responder cells that are among the first to encounter Mtb during transmission. Understanding the initial cellular transcriptional response of these cells, in the context of a cytotoxic Mtb aggregate infection, could be instrumental in predicting overall infection trajectory and reveal novel targets for intervention. Likewise, understanding mechanistic factors in the host-pathogen interaction, such as mycobacterial viability and phagosomal acidification, during an Mtb aggregate infection could reveal important determinants of infection outcome. Mtb aggregates have previously been found in resected lung tissue and shown to exacerbate infection outcomes in a rabbit model [147, 148]. The extent to which they are present in an *in vivo* human granulomatous environment, however, has not been quantified. Understanding the distribution and abundance of Mtb aggregates present in a granuloma may be informative in establishing whether this bacterial phenotype represents a relevant mechanism of pathogenesis. However, manually quantifying aggregated Mtb in scanned tissue slides is a time-consuming endeavour. Automated detection of Mtb bacilli in diagnostic sputum smears is widely employed [149-152], but solutions for Mtb detection in the more heterogenous context of tissue sections are less common. A reliable and more automated solution for detecting these bacilli is possible by training a CNN, as they are regularly applied to contemporary image classification tasks [129, 130].

In the current study I aim to investigate the macrophage response to Mtb aggregate infection and to assess the potential relevance of Mtb aggregates during human lung infection. The objectives are:

1. To understand the effect of Mtb multiplicity of infection and aggregation state on the macrophage transcriptional response (**Chapter 3**).

2. To investigate the effect of Mtb aggregate viability on macrophage death and acidification in response to infection with Mtb aggregates (**Chapter 3**).

3. To develop a custom image analysis pipeline to automate the quantification of Ziehl-Neelsen stained Mtb bacilli in human lung tissue sections (**Chapter 4**).

4. To quantify Mtb aggregates in TB positive human lungs and investigate the distribution of Mtb aggregates at pathological sites during *in vivo* infection (**Chapter 5**).

2 Chapter 2. Materials and methods

2.1 Ethical statement

Written informed consent was obtained for blood drawn from adult healthy volunteers (University of KwaZulu-Natal Institutional Review Board approval BE022/13). Written informed consent was obtained for lung sections from clinically indicated lung resections resulting from TB complications (University of KwaZulu-Natal Institutional Review Board approval BE019/13).

2.2 Macrophage culture

Peripheral blood mononuclear cells were isolated using density gradient centrifugation in Histopaque 1077 (Sigma-Aldrich, St Louis, MO). Purified CD14+ monocytes were obtained by positive selection using anti-CD14 microbeads (Miltenyi Biotec, San Diego, CA). For RNA-Seq experiments, CD14+ monocytes were seeded at 1 x 10^6 cells per well in non-tissue culture treated 35 mm 6-well plates. For time-lapse microscopy protocols, CD14+ monocytes were seeded at 0.2 x 10^6 cells per 0.01% fibronectin (Sigma-Aldrich) coated 35 mm glass bottom optical dishes (Mattek, Ashland, MA). Monocytes were differentiated in macrophage growth medium containing 1% each of HEPES, sodium pyruvate, L-glutamine, and non-essential amino acids, 10% human AB serum (Sigma-Aldrich), and 50 ng/ml GM-CSF (Peprotech, Rocky Hill, NJ) in RPMI. The cell culture medium was replaced one, three, and six days after plating.

2.3 Mtb culture and macrophage infection

Fluorescent H37Rv Mtb was derived by transforming the parental strain with a plasmid with mCherry under the smyc' promoter (generous gift from D. Russell). Mtb were maintained in Difco Middlebrook 7H9 medium enriched with oleic acid-albumin-dextrose catalase supplement (BD, Sparks, MD). Three days prior to macrophage infection, Mtb were grown in Tween 80-free media. On the day of infection, exponentially growing bacterial culture was pelleted at 2000g for 10 min, washed twice with 10 ml PBS, and large aggregates broken up by shaking with sterilized 2–4 mm glass beads for 30 s (bead beating). 10 ml of PBS was added, and large clumps were further excluded by allowing them to settle for 5 min. To generate single Mtb bacilli, the bacterial suspension was passed through a 5 µm filter syringe after aggregate preparation. The resulting singlet and aggregate Mtb suspensions were immediately used to infect MDMs. Mtb grown in media containing Tween 80 (Sigma-Aldrich) surfactant was grown in parallel with detergent free Mtb culture to monitor bacterial growth and calibrate macrophage infection using optical density. MDM were infected with 150µl Mtb aggregate suspension or 1000µl singlet suspension for 3 hours, washed with PBS to remove extracellular Mtb and incubated for a further 3 hours. Where heat-killing was required, the Mtb suspension was placed in a heating block for 20 minutes at 80^{o}C.

2.4 Isolation of infected macrophage populations for RNASeq

After infection (as above), MDMs were lifted from non-tissue culture treated plates, using 1ml of Accutase (Sigma-Aldrich) cell dissociation reagent per 35mm well, and transferred to FACS tubes. The cell viability stain Draq7 (BioStatus, Leicestershire, UK) was added at a concentration of 1:1000 per sample. Macrophage populations were separated into high and low Mtb infected populations using Mtb mCherry fluorescence levels (measured at 561nm) (Figure 3.1). Dead cells were removed based on DRAQ7 signal (633nm excitation). I estimated the number of bacilli per infected macrophage by comparing fluorescence distributions of Mtb in FACS data to the corresponding Mtb fluorescence distribution from confocal microscopy. Previous observations have shown a tight correspondence between bacterial colony forming units (CFU) and bacterial fluorescence (**Supplementary material Figure 8.1 adapted from Mahamed *et al.* [30])**. Infected MDMs, at 10 000 cells per tube, were sorted into Trizol (Thermo-Fischer, Massachusetts, USA) and snap frozen in a dry ice and 99 percent isopropanol slurry using a BD FACSAria III flow cytometer (BD, New Jersey, US).

2.5 RNASeq

Snap frozen MDM samples were stored at -80^{o}C before transport and sequencing. Sample cDNA libraries were prepared according to protocols established by John J. Trombetta *et al.* [153]. Zymogen Direct-Zol RNA Miniprep Kits (Zymo Research corporation, Irvine, CA) were used to extract RNA from cells frozen in Trizol, according to manufacturer instructions, followed by an RNAClean SPRI bead Cleanup (Beckman Coulter Life Sciences, Indianapolis, IND) and cDNA library prep using the Maxima H minus reverse transcriptase kit (Thermo-Fischer, Massachusetts, USA). The KAPA HiFi HotStart readymix (Sigma-Aldrich, St Louis, MO) was used for whole transcriptome amplification prior to adapter ligation, sequence pooling and sequencing on an Illumina NextSeq 500/550 instrument (Illumina, San Diego, CA) at the Shalek lab in the Broad Institute of MIT in Boston (Merkin Building, 415 Main St, Cambridge, MA 02142, United States). Transcripts were aligned to human reference hg19 and read count libraries generated using the RSEM software package [154]. Fastq sequence files were uploaded to the NCBI GEO (https://www.ncbi.nlm.nih.gov/geo/) under the accession number GSE173560.

2.6 Transcriptomics data analysis

Read count libraries for each of the 15 replicates per infection condition were generated at an average read depth of 4 per base [155] and processed using the DESeq2 package for the R programming platform (R foundation, Vienna, Austria) [122]. Metadata and read count matrices from each batch were concatenated into a single metadata and read count matrix prior to processing. For PCA count matrices were R-log normalized in DESeq2 and corrected for batch effects using the SVA ComBat package for R [156]. Read count matrices were RLog normalized and arranged in descending order by variance across treatment conditions, and the top 0.1% of variable genes plotted in a PCA to assess clustering. The DESeq2 negative binomial model was used to perform differential expression (DE) analysis, including blood donor as a factor. Genes identified in DE analysis were cross-referenced with RLog normalized, variance ordered lists to narrow gene candidate lists. Candidate genes were then tested for significant differences between infection conditions using a Hochberg corrected, non-parametric Mann-Whitney U-test. Ranked gene lists generated by DESeq2 DE analysis were used in gene set enrichment analysis (GSEA) using the Hallmark molecular signatures database [157] (https://www.gsea-msigdb.org/gsea/msigdb/index.jsp). Normalised enrichment scores were calculated for each of the comparisons and arbitrary cut-off values of $p < 0.001$ and FDR < 0.005 were used to identify significant functional regulatory categories between infection comparisons.

2.7 Cytokine Analysis

MDMs were isolated, differentiated, and infected with single or aggregated Mtb suspension, as above, and incubated for 3h before being washed with PBS to remove extracellular Mtb and incubated for a further 3h. Supernatant was collected, 0.2µm filtered and frozen prior to cytokine quantification. Cytokine levels were measured using a custom R&D Systems Luminex cytokine panel kit (R & D Systems, Minneapolis, MN), according to kit instructions, on a Biorad Bioplay 200 instrument (BioRad, California, US).

2.8 Microscopy

Macrophages and bacteria were imaged using an Andor (Andor, Belfast, UK) integrated Metamorph-controlled (Molecular Devices, Sunnyvale, CA) Nikon TiE motorized microscope (Nikon Corporation, Tokyo, Japan) with a 20x, 0.75 NA phase objective. Images were captured using an 888 EMCCD camera (Andor). Temperature and $C0_2$ were maintained at $37^{o}C$ and 5% using an environmental chamber (OKO Labs, Naples, Italy). During time-lapse protocols, images were captured once every ten minutes for the duration of the time-lapse. For each image acquisition, images were captured at wavelengths applicable to fluorophores used in the analysis and included transmitted light (phase contrast), 561nm (Red fluorescent protein), and 640nm (DRAQ7, lysotracker). Image analysis for fluorescence microscopy was conducted using custom written MATLAB (Mathworks, Massachusetts, USA) script. Single cells were manually segmented prior to fluorescent signal quantification. For each cell, fluorescent signal in each channel was quantified as pixel intensity.

2.9 Macrophage acidification

Single cell fluorescence data for lysotracker acidification was acquired at a single time point at 6 hours post infection using the confocal microscopy system described above. MDMs on fibronectin coated optical dishes were infected with 400µl Mtb aggregate suspension (**Materials and methods section 2.3**) and incubated for 3 hours before being washed with PBS to remove any cell free Mtb and incubated for a further 2 hours. 1 hour prior to microscopic image acquisition, 75nM Lysotracker (Thermo-Fischer, Massachusetts, USA) was added to wells. Image data was processed as above to determine pixel fluorescence intensity in each fluorescent channel per cell (**Materials and methods section 2.8)**. Macrophage surface area/volume (SA/V) model fit was to $3/r$ (a simplified expression for $spherical\ surface\ area/spherical\ volume$), where r was aggregate radius.

2.10 Combination staining of human lung

Human lung tissue was cut into 2 mm thick sections and picked on charged slides. Slides were baked at 56ºC for 15 minutes. Mounted sections were dewaxed in xylene followed by rinse in 100% ethanol and 1 change of SVR (95%). Slides were then washed under running water for 2 minutes followed by antigen retrieval via heat induced epitope retrieval (HIER) in Tris-sodium chloride (pH 6.0) for 30 minutes. Slides were then cooled for 15 minutes and rinsed under running water for 2 minutes. Endogenous peroxide activity was blocked using 3% hydrogen peroxide for 10 minutes at room temperature (RT). Slides were then washed in PBST and blocked with protein block (Novolink) for 5 min at RT. Sections were incubated with primary antibodies for CD68 (M0814-CD68-KP1, DAKO,1:3000) followed by washing and incubation with the polymer (Novolink) for 30 minutes at RT. Slides were then washed and stained with DAB for 5 minutes and washed under running water for 5 minutes.

For combination staining, slides were incubated with heated carbol fuchsin for 10 minutes and then washed in running tap water. 3% acid alcohol was applied to the slide to decolourize for 30 seconds or until sections appeared clear. Slides were then washed in running tap water for 2 minutes and where then counter stained with methylene blue. Slides were rinsed under running water, dehydrated, and mounted in Distyrene Plasticiser Xylene (DPX).

2.11 Image Processing

All image processing pipelines were developed using the MATLAB® programming platform.

2.11.1 Image scanning and input

Stained tissue sections were scanned to digital .ndpi format using a Nanozoomer 2.0 rs (Hamamatsu, Yokohama, Japan) slide scanner. RGB Images of resected lung tissue in .ndpi format were converted to .TIFF file types (without compression), using the NDPITools plugin for the ImageJ software package [158], to enable compatibility with MATLAB® (Mathworks, Massachusetts, USA) image processing functions. The resultant image files were imported into MATLAB® in smaller sections (tiles) to circumvent maximum array dimension constraints. This was calculated by finding the image array size (in pixels) along each dimension of the image and determining a divisor that resulted in a value closest to 5000 pixels. The full image was then divided into sections without remainder (**Code appendix section 8.2.1**).

2.11.2 Image pre-processing

Each layer of an input RGB image matrix was separated into individual grayscale matrices that contained only red (R), blue (B) or green (G) pixel intensity values. Additional cyan (C), magenta (M) and yellow (Y) grayscale intensity matrices were calculated from the R, G and B matrices using element wise matrix addition and subtraction. C was calculated by first adding B to G pixel intensity and then subtracting R pixel intensity, M by adding R to B and then subtracting G, and Y by adding R to G and then subtracting B (**Code appendix section 8.2.1**).

2.11.3 Binary masks

I generated a binary image mask, for each input image tile, to extract only the pixels that fell within the ZN colour profile. I generated 3 biased grayscale image matrices that had higher intensity values in pixels that contributed most to the ZN colour profile (**section 4.2.2,** R, G or B pixel values). This was done using element wise matrix subtraction. To upwardly bias M pixel intensities, I subtracted the G intensity matrix from the M matrix, for R I subtracted C from the R matrix and for B, I subtracted Y from the B matrix. I then removed background noise using the mean and standard deviation of pixel intensity for the individual biased matrices so that only higher intensity pixel values remained (**Code appendix section 8.2.1**). I created the final binary mask using element wise Boolean intersection between the 3 biased matrices. Pixels that were common to all 3 matrices were labelled as 1, and those that were not common in all 3 were labelled as 0.

2.11.4 Feature extraction

Binary masks (**Materials and methods section 2.11.3**) were used to select ZN-stained pixels in RGB images. Each binary mask was applied to the corresponding RGB image of origin to extract only pixels at positions where the binary matrix had a value of 1. Pixels selected in this way were then grouped by proximity. All pixels that were in direct contact with another pixel were considered to be a single object, with unique features, and allocated a unique identification number. Feature data for each object was then extracted from the RGB image (**Section 4, Figure 4.1**). These featured included the mean, median, mode, maximum, sum and standard deviation for R, G and B pixel intensities, as well as hue, saturation, value, pixel area, circularity, eccentricity, solidity, scale-invariant feature transform (SIFT) features and maximally stable extremal regions (MSER) features per object (**Code**

appendix section 8.2.1) [159, 160]. Feature data for objects and unique ID numbers were stored in a separate database for each scanned slide.

2.11.5 Manual object curation

I reduced the number of objects that did not match the ZN colour profile by manually removing them from the database for each slide. Mean R, G and B values for all objects were plotted on a 3D axis (**Section 4, Figure 4.3**). On the same set of axes, I plotted the mean R, G and B values of a separate and complementary database of objects that were derived from the same image tile. Complementary object databases were derived identically and in parallel to ZN object databases but using extracted pixels whose colour profiles were the opposite of ZN (**Materials and methods section 2.11.3**). This guided the coarse manual removal of a large number of objects that did not belong to the target distribution of ZN stained Mtb (ZN-Mtb), prior to further individual object validation (**Materials and methods section 2.11.6**).

2.11.6 Individual object validation

I validated individual ZN-Mtb objects generated by feature extraction (**Materials and methods section 2.11.4**) in the context of the immediately surrounding tissue. For each tile generated from tissue slides during image pre-processing (**Materials and methods section 2.11.2**) I generated a MATLAB® figure and overlaid the positions of the objects generated by feature extraction. For each tile I manually inspected all objects at these positions and removed those objects that were not ZN-Mtb from the database (**Code appendix section 8.2.3**). Additionally, aggregated or single Mtb classification was manually validated for a single slide. I calculated the mean area of manually validated single Mtb bacteria in pixels and used a conversion factor, stored in image metadata, to estimate bacterial size in microns. Mean single Mtb area was then used to estimate the number of bacteria in Mtb objects. I used the largest single bacterium as a cut-off value, after which Mtb objects were classified as aggregates. Datasets for each slide were then modified to include the manually validated labels for all objects. These datasets were saved for CNN training or graphing.

2.12 Convolutional neural network

All CNN functions were developed using the MATLAB® programming platform.

2.12.1 Initialization of CNN parameter weights

Code for the convolutional neural network was written based on methods (feedforward neural networks, backpropagation and gradient descent) collated, reviewed and developed by Jürgen Schmidthuber and others [146, 161], using the MATLAB® programming platform (Mathworks, Massachusetts, USA). A randomized matrix of parameters was generated for each layer of nodes in the neural network to provide a non-zero set of starting parameters for optimizing NN model cost according to equation [2] (**Code appendix section 8.2.14**).

$$\Theta^{j} = A \cdot 2\varepsilon - \varepsilon \qquad [2]$$

Where Θ^{j} is a randomized matrix of parameters for layer j of the NN, and A is a randomized $m \cdot n$ matrix with dimension m matching the number of nodes in the input layer and n matching the number of nodes in the output layer. ε is defined as in equation [3]

$$\varepsilon = \sqrt{6}/\sqrt{(m+n)} \qquad [3]$$

2.12.2 Forward propagation

Once parameter matrices were initialized, $m \cdot n$ training data matrices X (where m is number of training examples and n is the number of features) were fed into a forward propagation algorithm to generate a prediction for each training example using the randomized parameter matrices. For node activations in the first input layer j of the CNN, equation [4] was used to calculate the activations a^j .

$$a^j = g(\Theta^j X) \qquad [4]$$

Where $g(z)$ is the sigmoid function as defined in equation [5].

$$g(z) = \frac{1}{1+e^{-z}} \qquad [5]$$

For hidden layers (j) of the CNN or in the output hypothesis layer, activation is calculated as defined in equations [6] and [7] respectively (Code appendix **section 8.2.15**).

$$a^j = g(\Theta^j a^{j-1}) \qquad [6]$$

$$h_\Theta(x) = g(\Theta^j a^{j-1}) \qquad [7]$$

2.12.3 CNN cost function

Equation [8] was then used to determine the cost $J(\Theta)$ (cost J, given parameter matrices Θ) associated with a hypothesis prediction by the NN, given example i ($h_\Theta(x^i)$).

$$J(\Theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^i \log{(h_\Theta(x^i))_k} + (1 - y_k^i)\log{(1 - \left(h_\Theta(x^i)\right)_k}]$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl+1}(\Theta_{ji}^l)^2$$

[8]

Where m corresponds to number of training examples, i is the i^{th} training example and k is the k^{th} solution corresponding to training example i, such that y_k^i is the k^{th} solution corresponding to the training example x^i and $(h_\Theta(x^i))_k$ is the prediction h_Θ , given training example x^i and that corresponds to solution y_k. λ is a constant forming part of the regularization term in equation [8]. l is the number of layers in the neural network, and sl is the number of units s in network layer l. The regularization term is applied only so long as $j \neq 0$ (**Code appendix section 8.1.7**).

2.12.4 CNN Gradient descent and backpropagation

Gradient descent was used to adjust parameters for each layer of the NN and optimize the cost $J(\Theta)$, until convergence, as described in equation [9].

$$\Theta_{in}^{j} := \Theta_{in}^{j} - \alpha \frac{\partial}{\partial \Theta_{in}^{j}} J(\Theta) \tag{9}$$

α is a constant describing the learning rate of gradient descent. $\frac{\partial}{\partial \Theta_{in}^{j}} J(\Theta)$ is the partial derivative of the cost function $J(\Theta)$ with respect to Θ_{in}^{j}. Where Θ_{in}^{j} corresponds to the parameter in the i^{th} row and the n^{th} column from the parameter matrix in layer j. The partial derivative of $J(\Theta)$ with respect to parameter Θ_{in}^{j} was calculated using backpropagation as described by P Werbos [146], using equations [10], [11],[12] and [13].

$$\delta_j^l = a_j^l - y_j \tag{12}$$

$$\delta_j^l = (\Theta^l)^T \delta^{l+1} \cdot a^l \cdot (1 - a^l) \tag{11}$$

$$\Delta_{ij}^l := \Delta_{ij}^l + a_j^l \delta_j^{l+1} \tag{12}$$

$$\mathrm{D}_{ij}^l := \frac{1}{m} \Delta_{ij}^l + \lambda \Theta_{ij}^l \tag{13}$$

Where δ_j^l is the error of node j in layer l, and equation [9] is applied only to the output layer of the NN and equation [11] to all other layers except the first (input) layer, for which there is no error term. $(\Theta^l)^T$ denotes the transpose of the parameter matrix for layer l. Δ_{ij}^l is the cumulative error for all nodes in all layers. D_{ij}^l is equal to the partial derivative term $\frac{\partial}{\partial \Theta_{in}^{j}} J(\Theta)$ and is regularized so long as $j \neq 0$.

3 Chapter 3: The transcriptional response of macrophages infected with Mtb aggregates

3.1 Background

It has previously been shown that infection of macrophages with Mtb aggregates leads to an enhanced cell death response [30]. The frequency of macrophage death increased with increasing numbers of infecting Mtb, and time until cell death was inversely related to increasing numbers of infecting Mtb bacilli (**Introduction, Figure 1.7**). Interestingly, MDMs infected with single large Mtb aggregates died at a higher frequency than MDMs that were infected with a similar number of Mtb bacilli that were phagocytosed as multiple singles or smaller aggregates (**Introduction, Figure 1.8**). The type of cell death elicited in MDMs more closely resembled a necrotic cell death mechanism, such as pyroptosis, rather than apoptosis. And Mtb growing within dead MDMs showed higher growth rates than Mtb in extracellular media or live MDMs. Moreover, Mtb aggregates that killed MDMs could kill subsequent phagocytes that internalized the aggregates with greater efficiency (**Introduction, Figure 1.9**).

These observations prompted investigation of transcriptional regulatory events during the early cellular response to infection with Mtb aggregates. I wanted to identify the transcriptional signatures that characterized the increased cell death response seen in macrophages infected with Mtb aggregates. I designed experiments to characterize the transcriptional and functional response of macrophages to infection with Mtb aggregates. I investigated the differences between transcription profiles of MDMs infected with Mtb aggregates and MDMs infected with Mtb singlets at a similar MOI. I also investigated the transcriptional differences between macrophages infected with singlet Mtb bacilli at high MOI and low MOI.

Additionally, I wanted to investigate the effect of Mtb aggregates on macrophage phagosomal acidification. Phagosome acidification is an important intracellular signalling event in response to phagocytosed pathogens (**Introduction section 1.3.2-1.3.3**). Acidification facilitates the liberation of bacterial ligands by facilitating the activity of hydrolytic enzymes. PAMPs are then free to bind to TLRs which form part of a cellular signalling cascade in response to invading pathogens. Mtb is known to inhibit the phagosome acidification process. I wanted to determine if differences in macrophage phagosomal acidification could be caused by differences in the physical sizes of Mtb aggregates, and therefore whether differences in the cellular response to Mtb aggregates might be mediated by acidification that depended on aggregate size. I also wanted to investigate whether Mtb aggregates had to be alive to elicit any changes in the host death response, and whether bacterial viability had an effect on the macrophage acidification response. I set up experiments to quantify and model MDM cell death and acidification in response to infection with Mtb aggregates using time-lapse and single time point confocal fluorescence microscopy.

Research questions:

- Do macrophages respond differently to infection with Mtb aggregates, compared to infection with Mtb singles or multiple singles, during early transcriptional responses?
- Is macrophage acidification related to Mtb aggregate size during phagocytosis?
- Do Mtb aggregates need to be alive to elicit macrophage death?

Hypotheses:

- Mtb aggregate-infected macrophages have a unique transcriptional signature relative to Mtb singles, or multiple singles, during the early transcriptional response to infection.
- Macrophage acidification is dependant on Mtb aggregate size during phagocytosis
- Mtb aggregates must be alive in order to elicit cell death in infected macrophages

Objectives:

- Infect MDMs with Mtb aggregates or Mtb singles, sort macrophages by bacterial MOI using bacterial fluorescence and RNAseq the resultant populations
- Check for broad changes in gene expression between Mtb aggregate infected, Mtb single infected, and multiple single Mtb infected MDMs using clustering analysis
- Refine transcriptional results using differential expression analysis and identify single gene candidates that differ between infection conditions
- Identify functional changes, or functionally similar groups of genes, between infection conditions using Gene Set Enrichment Analysis.
- Infect macrophages with Mtb aggregates in the presence of an acidification reporter dye and quantify using confocal fluorescence microscopy
- Infect macrophages with live or heat-killed Mtb aggregates and quantify macrophage death using membrane permeability dye during time-lapse confocal fluorescence micrsocopy

3.2 Results

3.2.1 Phagocytosis of fluorescent H37Rv by macrophages enables FACS of infected macrophages by MOI

I sorted infected MDMs using the fluorescence of internalized mCherry fluorescent H37Rv Mtb bacilli. I infected each of 5 donor MDMs (3 repeats per infection condition, n = 60) with mCherry fluorescent expressing Mtb prepared as either single or aggregated bacteria (**Materials and methods, section 2.2, 2.3 and 2.4**). Infected MDMs were incubated for three hours, lifted from culture vessels, single cell sorted into the required populations and subsequently snap-frozen prior to transport and RNAseq (**Materials and methods, section 2.4 and 2.5**). MDMs were sorted into populations based on fluorescence of internalized Mtb (**Figure 3.1**). Sorted populations were labelled “Uninfected” for MDMs with no internal Mtb, “Single Mtb” for MDMs infected with disaggregated Mtb and a low amount of bacterial fluorescence (Low MOI gate, middle panel **Figure 3.1**), “Multiple Mtb” for MDMs infected with disaggregated Mtb and high Mtb fluorescence (High MOI gate, middle panel **Figure 3.1**) and “Aggregated Mtb” for MDMs infected with aggregated Mtb and high Mtb fluorescence (equal to fluorescence of the “Multiple Mtb” group, Left panel **Figure 3.1**). For each donor, I sorted 120 000 cells across 3 repeats and 4 treatments, for a total of 600 000 cells (10 000 cells per repeat). Frozen MDM populations were then sent for library prep and transcript sequencing to collaborators at the Shalek lab at MIT in Boston (**Materials and methods, section 2.5**). I analysed the resultant read count data.

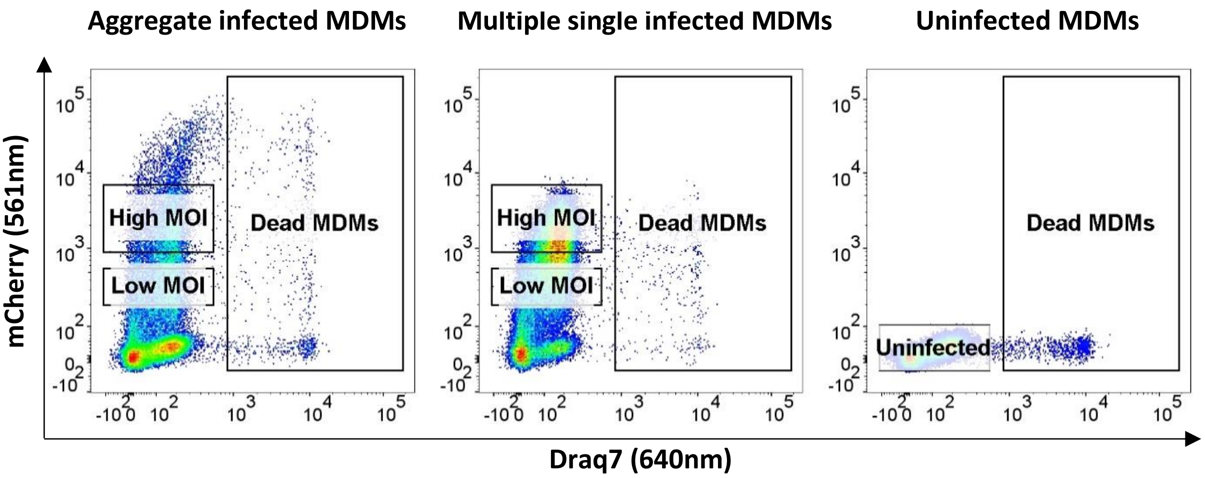


Figure 3.1: Macrophages were infected with aggregated or singlet Mtb and sorted by number of infecting bacilli

MDMs were infected with aggregated Mtb (left panel), disaggregated Mtb (middle panel) or uninfected (right panel) and sorted into populations based on quantity of infecting bacteria (bacterial fluorescence – y axis, mCherry). MDMs in the High MOI gate of aggregate infection were labelled “Aggregated Mtb”, MDMs in the High MOI gate of multiple single Mtb infection, and matching the MOI of aggregate infection, were labelled “Multiple single Mtb”, MDMs in the Low MOI gate of multiple single infection were labelled “Single Mtb”, and uninfected MDMs were labelled “Uninfected”. Dead MDMs were excluded using the cell permeability dye Draq 7 (x axis).

3.2.2 Transcriptional profiles of Mtb aggregate-infected macrophages cluster when using most variable genes

I used a PCA to cluster read counts and check for trends in expression between infection conditions. Before performing PCA analyses I applied the

DESeq2 RLog normalization function on the data to minimize disproportionate effect of genes with very low or very high read counts [122]. I also corrected for batch effects using the SVA ComBat package for R. I initially included all genes in the PCA analysis to see if samples clustered by infectious condition. I then reduced the number of genes included in the PCA by removing those genes which had the lowest variance between infection conditions. Low variance genes were removed in $\log_{10}$ orders of magnitude. I first used 100% of all genes (23686 genes) to construct the PCA, then the top 10% of most variable genes (2369 genes), then the top 1% of most variable genes (237 genes) and finally, the top 0.1% of most variable genes (24 genes) (**Figure 3.2**).

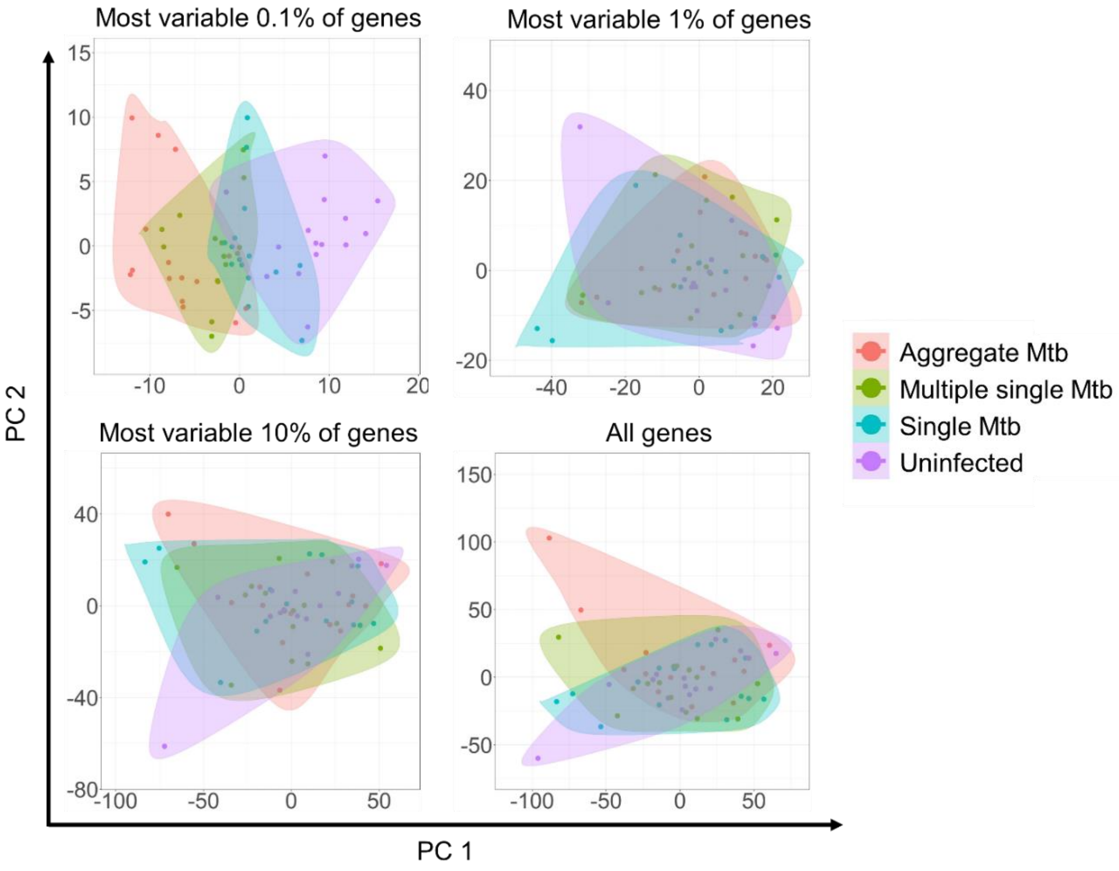


Figure 3.2 Transcription profiles clustered in a PCA using the most variable genes. Adapted w/o permission from Rodel *et al.* 2021 [162]

Samples clustered by infection condition for PCA utilising the top 0.1% of most variable genes. Clustering on this limited gene set showed aggregate infected MDMs (salmon) farthest removed from uninfected MDMs (purple), followed by multiple Mtb (green) and single Mtb (blue) respectively. PCAs using 1%, 10% or all genes did not show any obvious clustering.

This allowed us to visualize clustering and identify genes that contributed the most to separation in the PCA space. There was no obvious clustering along the first two principal component axes when using all genes, the top 10% most variable genes or the top 1% of most variable genes (**Figure 3.2**). However, when using most variable 0.1% to construct the PCA, the clusters separated by infection condition along the first principal component (**Figure 3.2, Adapted from Rodel *et al.* [162]**). Aggregated Mtb infection separated farthest away from the uninfected condition, followed by multiple single Mtb and then single Mtb infection respectively (**Figure 3.2**). I plotted the percentage contribution of each principal component for the PCA constructed using the most variable 0.1% of genes. Principal component 1 comprised 22% of the variation relative to all other principal components, followed by 9% variance contributed by principal component 2 (**Figure 3.3**). I finally plotted the percentage contribution of each gene to the principal component space. Genes that had the highest percentage contribution to separation along the first principal component were involved in the inflammatory response. Such genes included TNF, IL8, CCL4, IL1β, CXCL2 and CXCL3 (**Figure 3.4, Adapted from Rodel *et al.* [162]**).

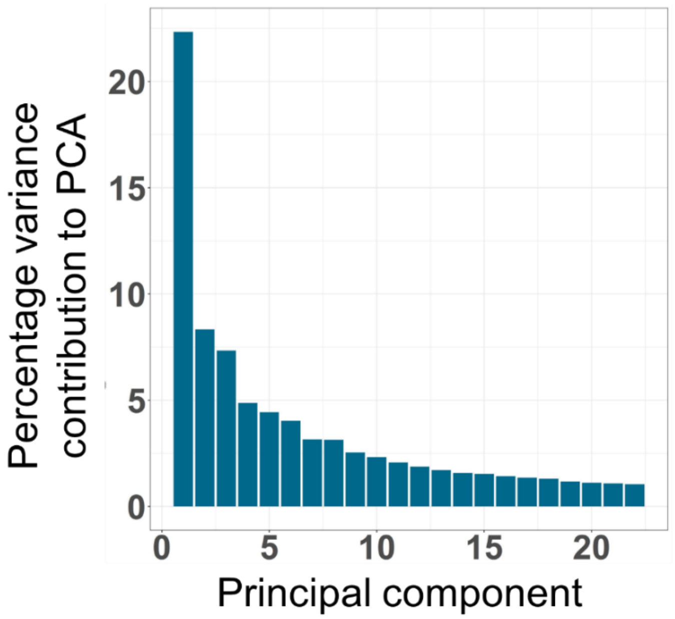


Figure 3.3 Principal component percentage contributions to variance of PCA using 0.1% of most variable genes PCA

Principal component 1 contributed 22% variability to the PCA in which there was clustering of infection conditions. Principal component 2 contributed 9% variability to the same PCA.

Genes that contributed the most to separation on the second principal component were involved in functions such as TNF-α upregulation and lymphocyte activation. These included IL32, CD69 and LCK.

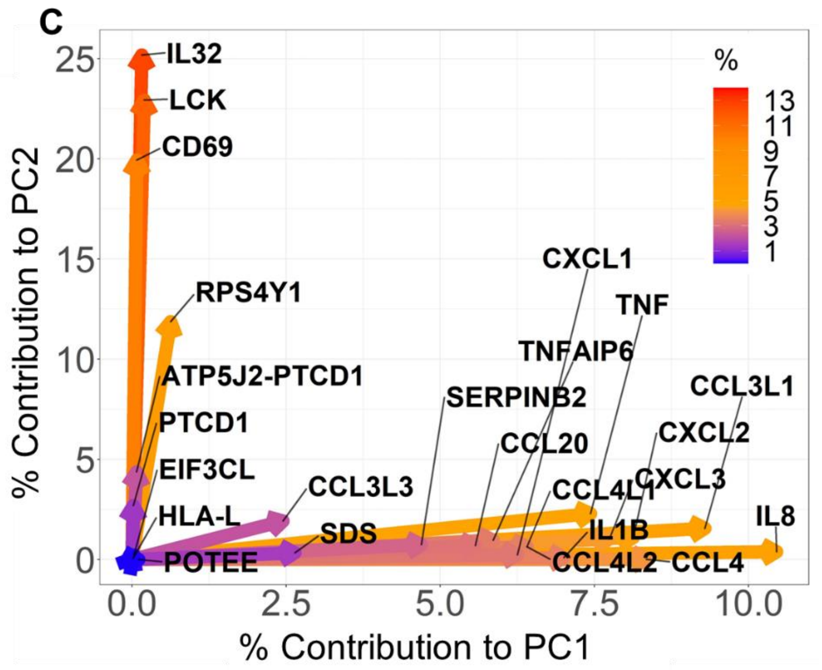


Figure 3.4 Individual percentage gene contributions to PCA constructed using 0.1% most variable genes. Adapted w/o permission from Rodel *et al.* 2021 [162]

Genes contributing to PC1 were primarily involved in the inflammatory response. Scale bar and colour gradient indicates percentage contribution to variation of the PCA.

3.2.3 Aggregated Mtb elicits a widespread transcriptional response in infected macrophages

DE analysis showed that aggregated Mtb elicited the significant differential expression of a higher number of genes in MDMs compared to multiple single or single Mtb infected MDMs, relative to uninfected macrophages. I tested for differential expression between infection conditions using the DESeq2 package for the R programming platform. I compared each infection condition to every other infection condition to generate comprehensive lists of significantly differentially expressed genes (**Supplementary material table**

8.1). I used the gene lists for each infection condition, compared to uninfected macrophages, to generate a Venn diagram showing the number of differentially regulated genes in each condition (**Figure 3.5, Adapted from Rodel *et al.* [162]**). Aggregate infected MDMs had the highest number of differentially regulated genes relative to uninfected MDMs at 160. 65 of these genes were shared with multiple Mtb infected MDMs, 37 were shared with single infected MDMs, and 34 were common to all infection types. Multiple Mtb infected MDMs had the second highest number of differentially regulated genes at 85, followed by single infected MDMs with 52 genes. However, aggregate infected MDMs had the highest number of uniquely regulated genes at 92 as compared to only 16 unique genes in multiple single Mtb infection and 11 genes in single Mtb infection.

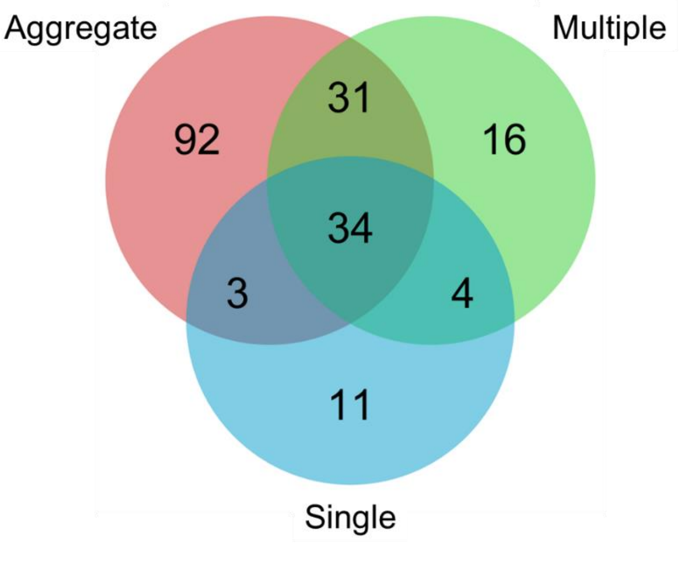


Figure 3.5 Mtb aggregate infection differentially regulated the highest number of unique genes in MDMs. Adapted w/o permission from Rodel *et al.* 2021 [162]

Venn diagram showing the number of differentially regulated genes in single (blue), multiple (green) and aggregate Mtb (salmon) infected macrophages relative to uninfected macrophages as identified by DESeq2 differential expression analysis (adjusted p value < 0.1).

3.2.4 Infected macrophages show a transcriptional response that is dependent on Mtb aggregation state and MOI

Next, I visualised single gene expression patterns between infection conditions. I used DE gene lists resulting from comparisons between infection conditions and plotted $\log_2$ fold change in gene expression against gene read count. I also highlighted those genes with highly significantly differences and labelled the top 10 differentially expressed genes in each comparison (**Figure 3.5**). Amongst the top 10 significantly regulated genes that were commonly regulated between all MDM infection states, relative to uninfected MDMs, were SERPINB2, TNFAIP6, IL1β, IL8, CCL4L1, CCL4L2 and SOD2 (**Figure 3.6, top 3 panels**). The majority of differential expression in these 3 comparisons was due to upregulation, relative to uninfected MDMs. Aggregate Mtb infection had the highest number of highly significantly regulated genes, at 48 genes (adjusted p-value < 0.0001). 10 of these genes had a $\log_2$ fold change greater than 5. This was followed by multiple Mtb infection with 26 genes (adjusted p-value < 0.0001), and 2 genes with $\log_2$ fold change greater than 5 and finally single Mtb infection with 16 genes (adjusted p-value < 0.0001) and a single gene with $\log_2$ fold change greater than 5 (**Figure 3.6 top row of panels**). Aggregate infection had more significantly differentially regulated genes, relative to single infection, than multiple Mtb infection compared to single Mtb infection (**Figure 3.6 bottom left and middle panel**). Only TNF-α was significantly upregulated in multiple infection relative to single Mtb infection (**Figure 3.6 bottom left panel**). However aggregate infection had a total of 54 significantly differentially regulated genes when compared directly to single Mtb infection, 7 of which had an adjusted p-value < 0.0001, and included the genes TNF, CCL4, IER3 and HSPA1A (**Figure 3.6 bottom middle panel**). Aggregate infection did not show highly significant (p < 0.0001) differences in gene expression when compared directly to multiple Mtb infection (**Figure 3.6 bottom right panel**). But genes that were significantly differentially regulated in this comparison included HSPA1A and IER3.

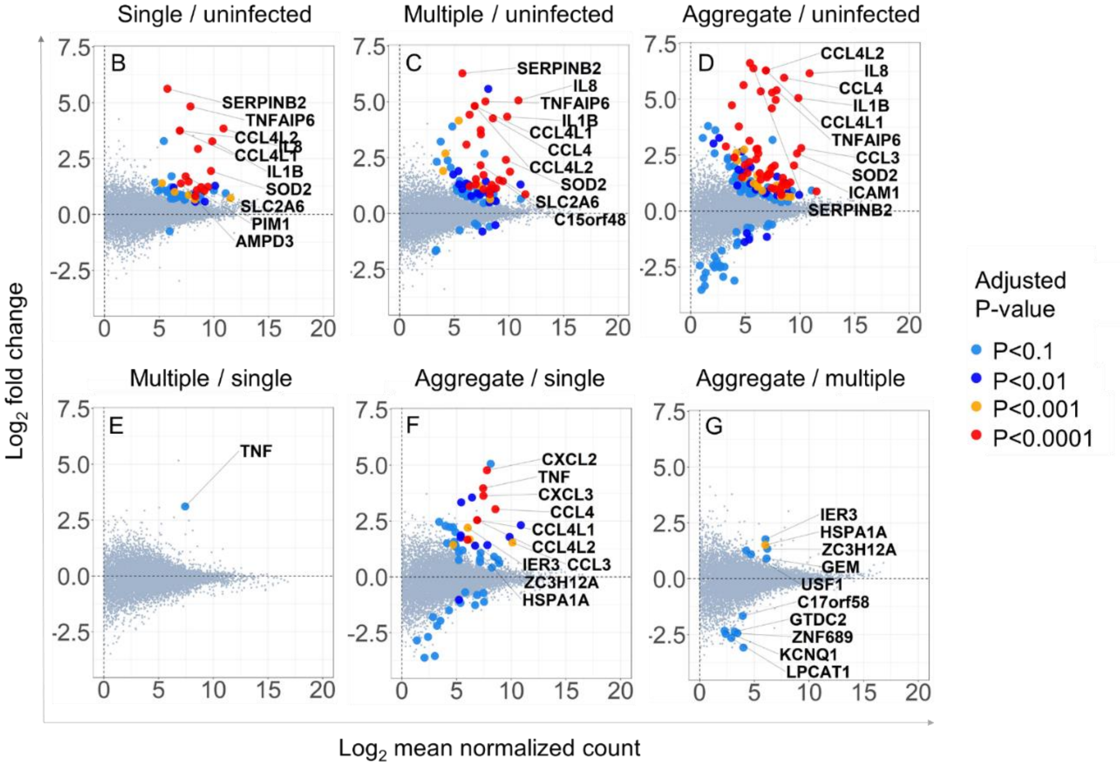


Figure 3.6 Single gene level expression patterns in MDMs were dependent on MOI and aggregation state

Aggregate infected MDMs had the highest number of significantly regulated genes relative to uninfected MDMs (light blue circles = p-value <0.1, dark blue circles = p-value <0.01, yellow circles = p-value <0.001, red circles = p-value <0.0001 and small grey circles = not significant). The top 10 genes differentially regulated in each comparison are labelled. Y axis is $\log_2$ fold change and X axis is $\log_2$ mean normalized read count. Values above perforated black line indicate upregulation and below indicates downregulation.

3.2.5 Single gene expression most often peaks in aggregated Mtb infection of MDMs

I generated an expression heatmap across infection conditions using a refined set of genes (**Figure 3.7, Adapted from Rodel *et al.* [162]**). I cross referenced the DE gene lists generated by DESeq2 analysis with each other and with the top 1% of most variable genes identified in the PCA analysis (**Section 3.2.2**). This resulted in a list of 21 genes including: IL1β, TNF, IL8, IL6, CCL4, CXCL2, CXCL3, IER3, SERPINB2, and TNFAIP6. Functions associated with these genes include TNF-α response, inflammation, neutrophil chemotaxis, and regulation of apoptosis [163-167] (**Figure 3.7**). Infection conditions and genes were hierarchically clustered by Euclidean distance. Data arranged so that there was a trend showing an increase in gene expression levels from uninfected, to singly infected, to multiply infected to aggregate infected (**Figure 3.7**). An exception to this trend was found in IL7R, which increased in expression from uninfected to singly infected, to multiply infected and then decreased again in the aggregate infection condition. UHRF1 also did not follow the predominant data trend and had the lowest expression in aggregate infection, with similar levels of expression in all other conditions.

3.2.6 TNF-α expression is dependent on Mtb aggregation state and MOI in infected MDMs

I next assessed whether the refined list of individual genes were significantly differentially regulated between the infection conditions using multiple comparisons. I tested for differences between uninfected and single infected, single infected and multiple infected and multiple infected and aggregate infected for each gene, using a non-parametric pairwise comparison corrected for multiple comparisons (**Materials and methods, section 2.6**). Of the original 21 genes identified, 14 were significantly different in at least one pairwise comparison (**Figure 3.8, Adapted from Rodel *et al.* [162]**). Two main patterns were evident in these comparisons. The first pattern showed significant differences in expression regardless of number of infecting bacteria. IL8, IL6, CCL4, TNFAIP6, IL1β, CCL4L1, SERPINB2, CCL20, CCL4L2, CXCL3, ADORA2A and UHRF1 followed this trend. The other expression pattern was dependent on bacterial number or aggregation state. TNF and IER3 followed this trend.

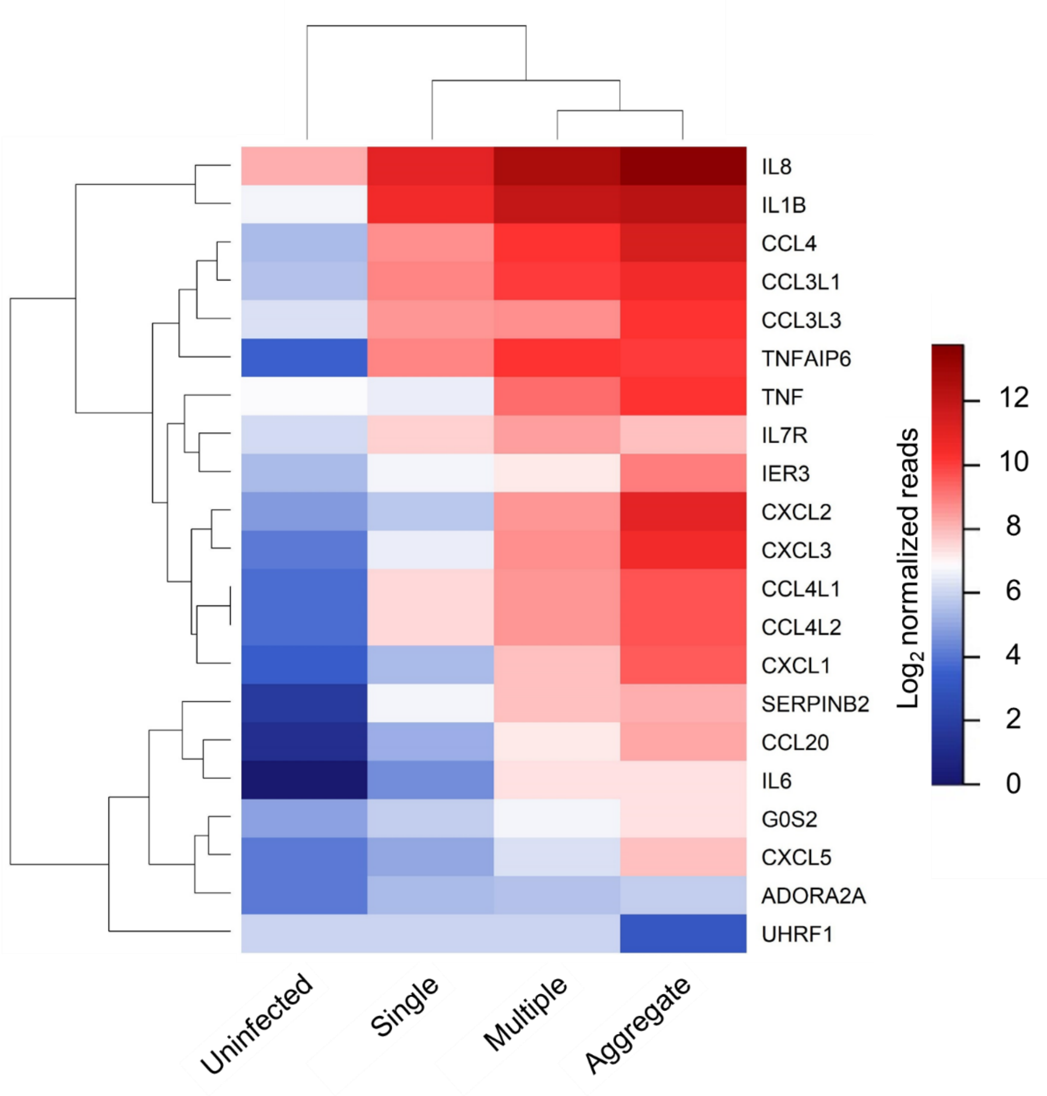


Figure 3.7 Gene expression for a refined list of genes most often peaked in aggregate infection of MDMs. Adapted w/o permission from Rodel *et al.* 2021 [162]

When using a refined set of 21 genes the infection conditions clustered, by Euclidean distance, so that aggregate infection of MDMs had the most upregulated gene expression levels, followed by multiple single Mtb infection, single Mtb infection and uninfected Mtb respectively. Log_2 normalized read counts are indicated by colour key.

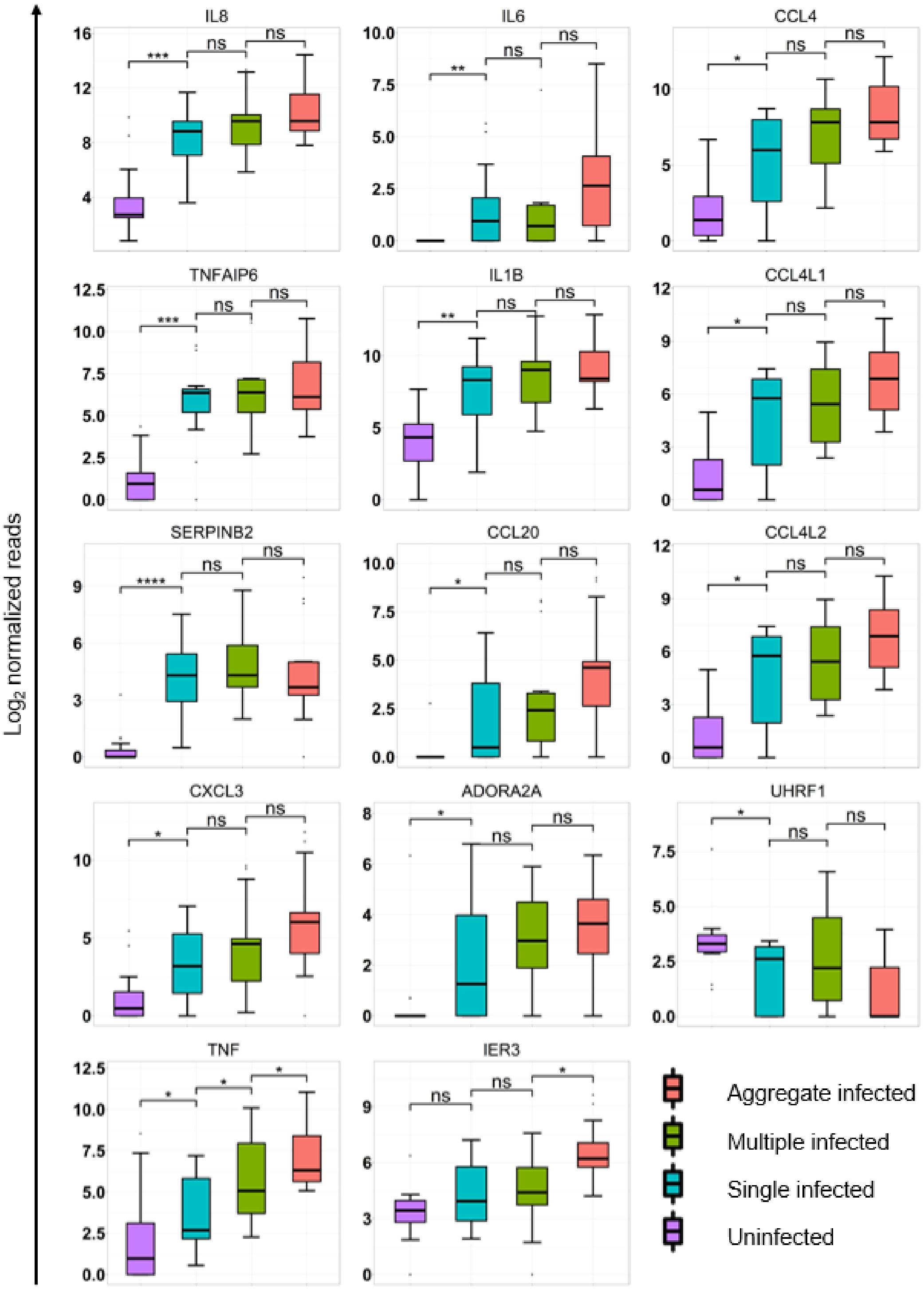


Figure 3.8: Aggregation state alters macrophage transcriptional response at the single gene level. Adapted w/o permission from Rodel *et al.* 2021 [162]

Box plots of median and interquartile range for read counts from 15 independent infections of MDM from 5 blood donors. Shown are expression levels as log transformed read counts in uninfected, single infected, multiple infected and aggregate infected macrophages. p-values are ns = not significant; * < 0.01; ** < 0.001; *** < 0.0001; **** < 0.00001; as determined by Mann-Whitney U non-parametric test with Hochberg multiple comparison correction.

I validated transcriptional results using cytokine panels. I infected MDMs with either aggregated or single Mtb, incubated the cells for 3 hours, and harvested culture supernatants for cytokine profiling (**Materials and methods, section 2.7**). I then compared the cytokine levels to read counts obtained for RNAseq (**Figure 3.9, Adapted from Rodel *et al.* [162]**). TNF-α showed a graded cytokine response to infection, increasing in the single Mtb infection condition (relative to uninfected MDMs) and upregulated further in the aggregate infection condition (**Figure 3.9 A**). Similar trends were observed in the RNAseq results for TNF-α (**Figure 3.9 B**). IL8 and IL6, key mediators of inflammation,

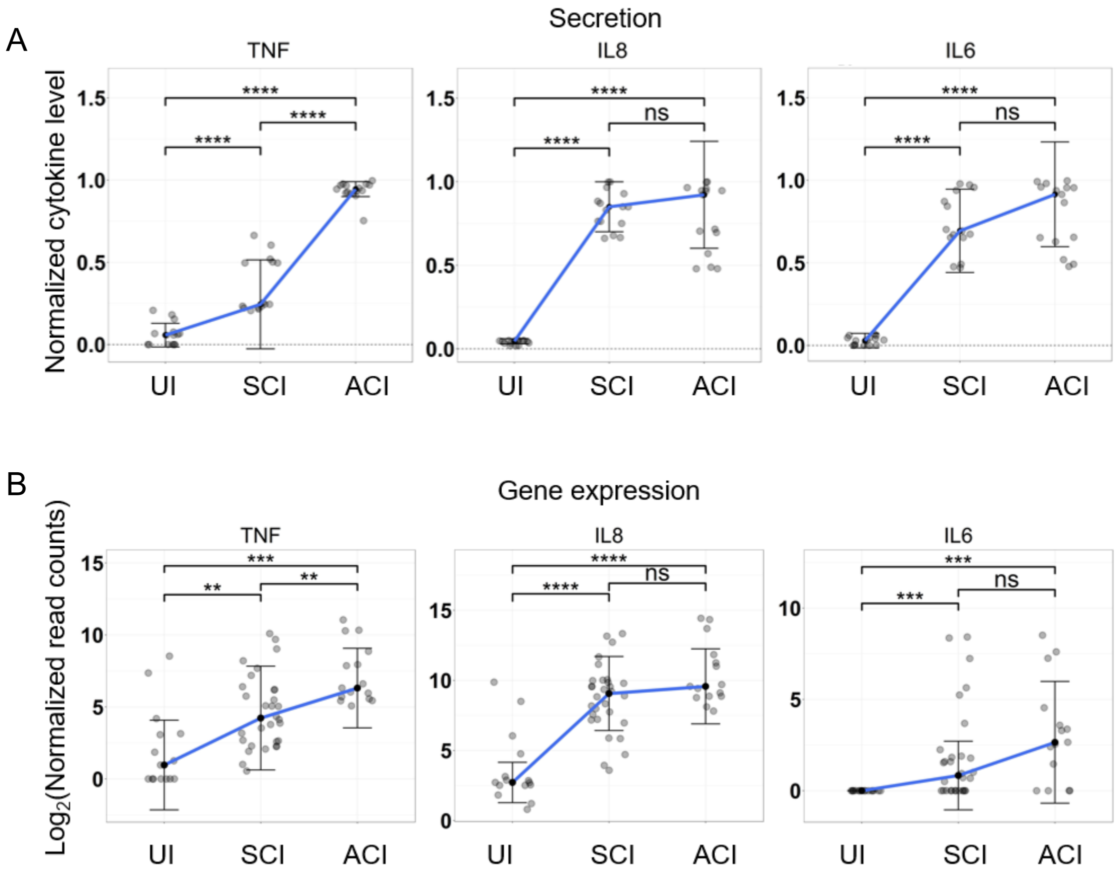


Figure 3.9: Transcriptional regulation and secretion of TNFα and downstream genes in aggregated and single Mtb infection conditions. Adapted w/o permission from Rodel *et al.* 2021 [162]

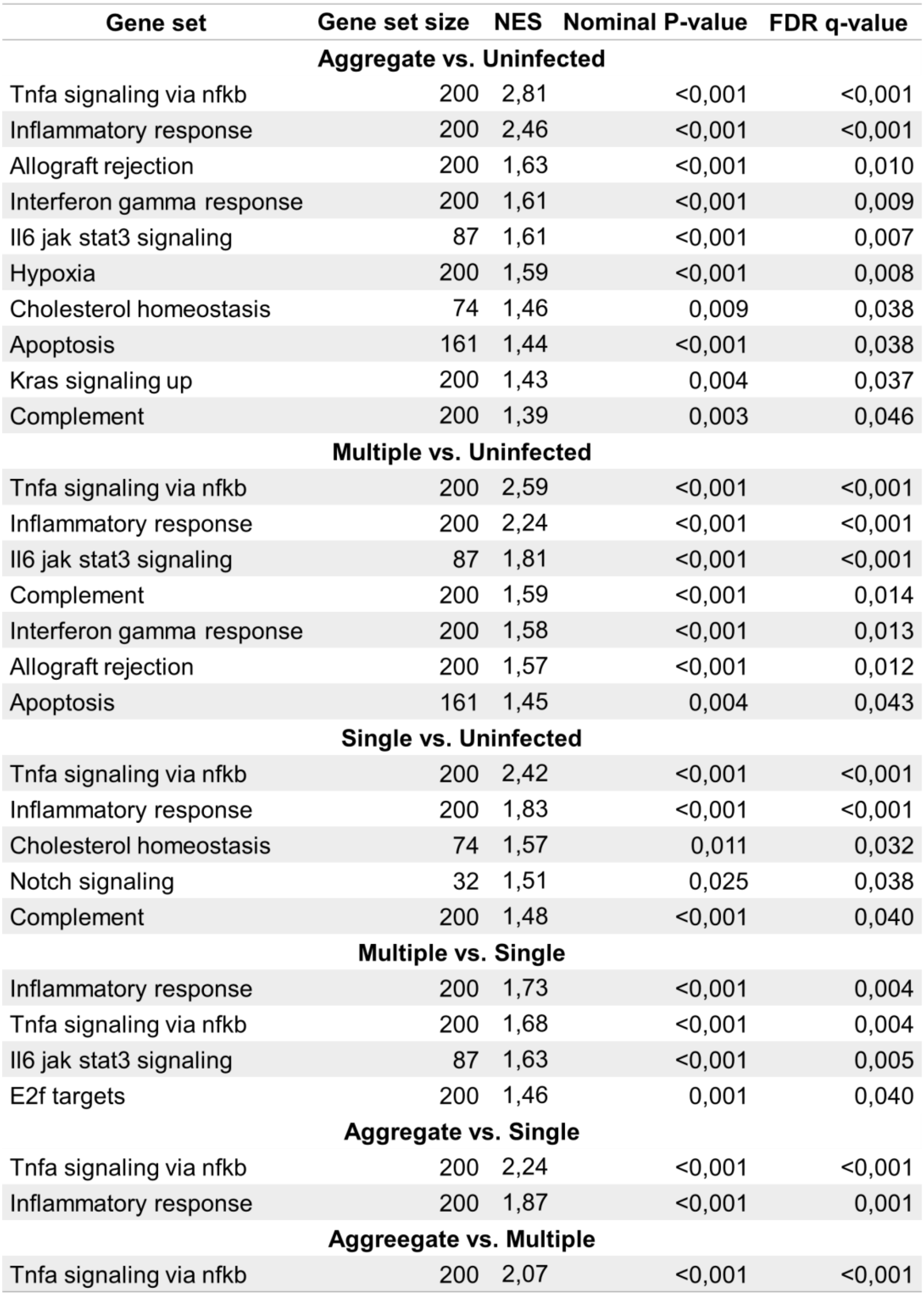
Cytokine secretion, normalized to maximum per target cytokine (A), or Log$_2$ normalized transcripts 3 hours post-Mtb infection (B). MDM were either uninfected (UI), infected with single Mtb culture (SCI) or with aggregated Mtb culture (ACI). Shown are median and IQR of the transcriptional or cytokine. p-values are * < 0.01; ** < 0.001; *** < 0.0001; **** < 0.00001 as determined by Mann-Whitney U test with Bonferroni multiple comparison correction.

showed indiscriminate upregulation in the presence of Mtb for cytokine profiles, regardless of whether the inoculum was single or aggregated bacteria (**Figure 3.9 A**). The same was true for the corresponding RNAseq read count data for the (**Figure 3.9 B**).

3.2.7 TNF-α signalling and inflammation are key functional pathways regulated in response to Mtb aggregation state and MOI

I used Gene Set Enrichment Analysis (GSEA) to identify the primary biological functions of differentially expressed genes. Ranked gene lists that were generated by differential expression analysis in DESeq2 were used to generate Normalized Enrichment Scores (NES) for functional groups of genes in GSEA using the Hallmark molecular signatures database (**Materials and methods, section 2.6**). I set the cut-off for significant functional regulation at nominal p-value < 0.001 and false discovery rate (FDR) < 0.005. The aggregate versus uninfected comparison had the highest number of differentially regulated functional categories at 10 (**Table 3.1**), followed by multiple versus uninfected with 7, single versus uninfected with 5, multiple versus single with 4, aggregate versus single with 2 and finally aggregate versus multiple with 1 functional category “tnfα signalling via NF-κB”. The “tnfα signalling via NF-κB” category was found within the top 2 differentially regulated categories in each of the comparisons, along with “inflammatory response” and had the highest NES in each comparison except the multiple vs. single condition in which “inflammatory response” had the highest NES (**Table 3.1, Adapted from Rodel *et al.* [162]**). The NES score for “tnfα signalling via NF-κB”, peaked in the aggregate versus uninfected comparison at 2.81 followed, in descending order, by multiple versus uninfected, single versus uninfected, aggregate versus single, aggregate versus multiple and multiple versus single at, 2.59, 2.42, 2.24, 2.07 and 1.68 respectively. (**Figure 3.10 A, Table 3.1, Adapted from Rodel *et al.* [162]**). Similarly, the descending order of NES score for infection comparisons in the “inflammatory response” category was aggregate vs. uninfected, multiple vs. uninfected, aggregate versus single, single vs. uninfected, and finally multiple versus single at 2.46, 2.24, 1.87 ,1.83 and 1.73 respectively (**Figure 3.10 B, Table 3.1**).

Table 3.1: GSEA differentially regulated gene sets between infection conditions at Nominal p-value and FDR<0.05. Adapted w/o permission from Rodel *et al.* 2021 [162]



Gene set	Gene set size	NES	Nominal P-value	FDR q-value
Aggregate vs. Uninfected				
Tnfa signaling via nfkb	200	2,81	<0,001	<0,001
Inflammatory response	200	2,46	<0,001	<0,001
Allograft rejection	200	1,63	<0,001	0,010
Interferon gamma response	200	1,61	<0,001	0,009
Il6 jak stat3 signaling	87	1,61	<0,001	0,007
Hypoxia	200	1,59	<0,001	0,008
Cholesterol homeostasis	74	1,46	0,009	0,038
Apoptosis	161	1,44	<0,001	0,038
Kras signaling up	200	1,43	0,004	0,037
Complement	200	1,39	0,003	0,046
Multiple vs. Uninfected				
Tnfa signaling via nfkb	200	2,59	<0,001	<0,001
Inflammatory response	200	2,24	<0,001	<0,001
Il6 jak stat3 signaling	87	1,81	<0,001	<0,001
Complement	200	1,59	<0,001	0,014
Interferon gamma response	200	1,58	<0,001	0,013
Allograft rejection	200	1,57	<0,001	0,012
Apoptosis	161	1,45	0,004	0,043
Single vs. Uninfected				
Tnfa signaling via nfkb	200	2,42	<0,001	<0,001
Inflammatory response	200	1,83	<0,001	<0,001
Cholesterol homeostasis	74	1,57	0,011	0,032
Notch signaling	32	1,51	0,025	0,038
Complement	200	1,48	<0,001	0,040
Multiple vs. Single				
Inflammatory response	200	1,73	<0,001	0,004
Tnfa signaling via nfkb	200	1,68	<0,001	0,004
Il6 jak stat3 signaling	87	1,63	<0,001	0,005
E2f targets	200	1,46	0,001	0,040
Aggregate vs. Single				
Tnfa signaling via nfkb	200	2,24	<0,001	<0,001
Inflammatory response	200	1,87	<0,001	0,001
Aggreegate vs. Multiple				
Tnfa signaling via nfkb	200	2,07	<0,001	<0,001

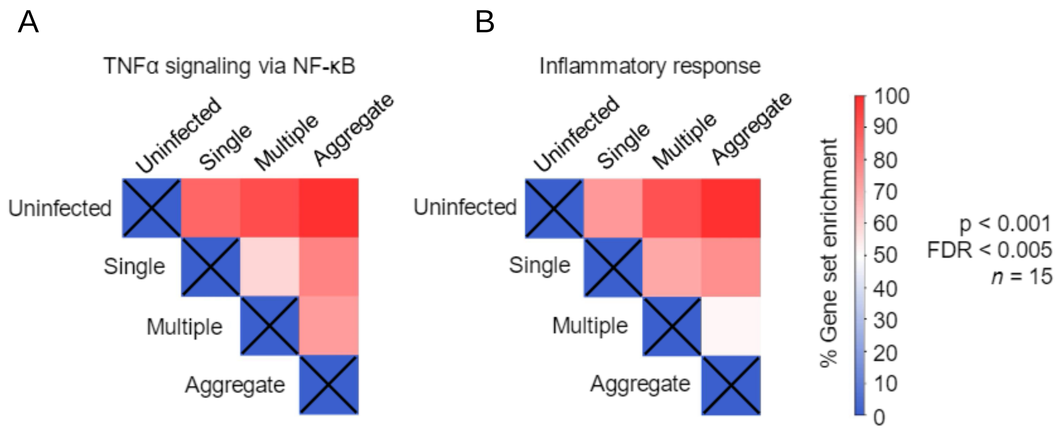


Figure 3.10: TNF-α signalling and inflammation are differentially regulated between infection conditions and peak in Mtb aggregate infection. Adapted w/o permission from Rodel *et al.* 2021 [162]

Normalised enrichment score (NES), expressed as percentage of maximum enrichment for the gene sets defined as ”tnfα signalling via NF-κB” (A), and ”Inflammatory response” (B). Enrichment scores were calculated for all treatment comparisons and were significantly different at Nominal p<0.001 and FDR<0.005, with the exception of the aggregate to multiple comparison for the ”inflammatory response” gene set (where p<0.05 and FDR = 0.24).

3.2.8 Large Mtb aggregates elicit lower macrophage acidification per bacillus

I measured macrophage acidification in response to infection with Mtb aggregates. I infected MDMs from 3 donors with mCherry expressing fluorescent Mtb aggregates and incubated with Lysotracker, a fluorescent acidification reporter. I then imaged infection using confocal fluorescence microscopy (**Figure 3.11 A, Adapted from Rodel *et al.* [162]**), at a single early time point 3h after infection, and quantified the Mtb and MDM acidification fluorescent signal within individual MDMs using custom MATLAB® image analysis scripts (**Materials and methods, section 2.9**). I did the same for MDMs infected with heat-killed fluorescent Mtb. A total of 3357 individual cells were analysed this way. I plotted lysotracker fluorescence against Mtb area (in pixels) and found that acidification increased with increasing Mtb aggregate area, with a general linear model fit of $R^2 = 0.63$ (**Figure 3.11 B, left panel**), I then took the ratio of Lysotracker signal to Mtb fluorescent signal and plotted it against Mtb area. I found that, as the area of an aggregate increases, the ratio of Lysotracker signal to Mtb signal decreases. I also fitted a SA/V model (spherical) to the data and attained a fit of $R^2 = 0.25$ (**Figure 3.11 B, right panel**).

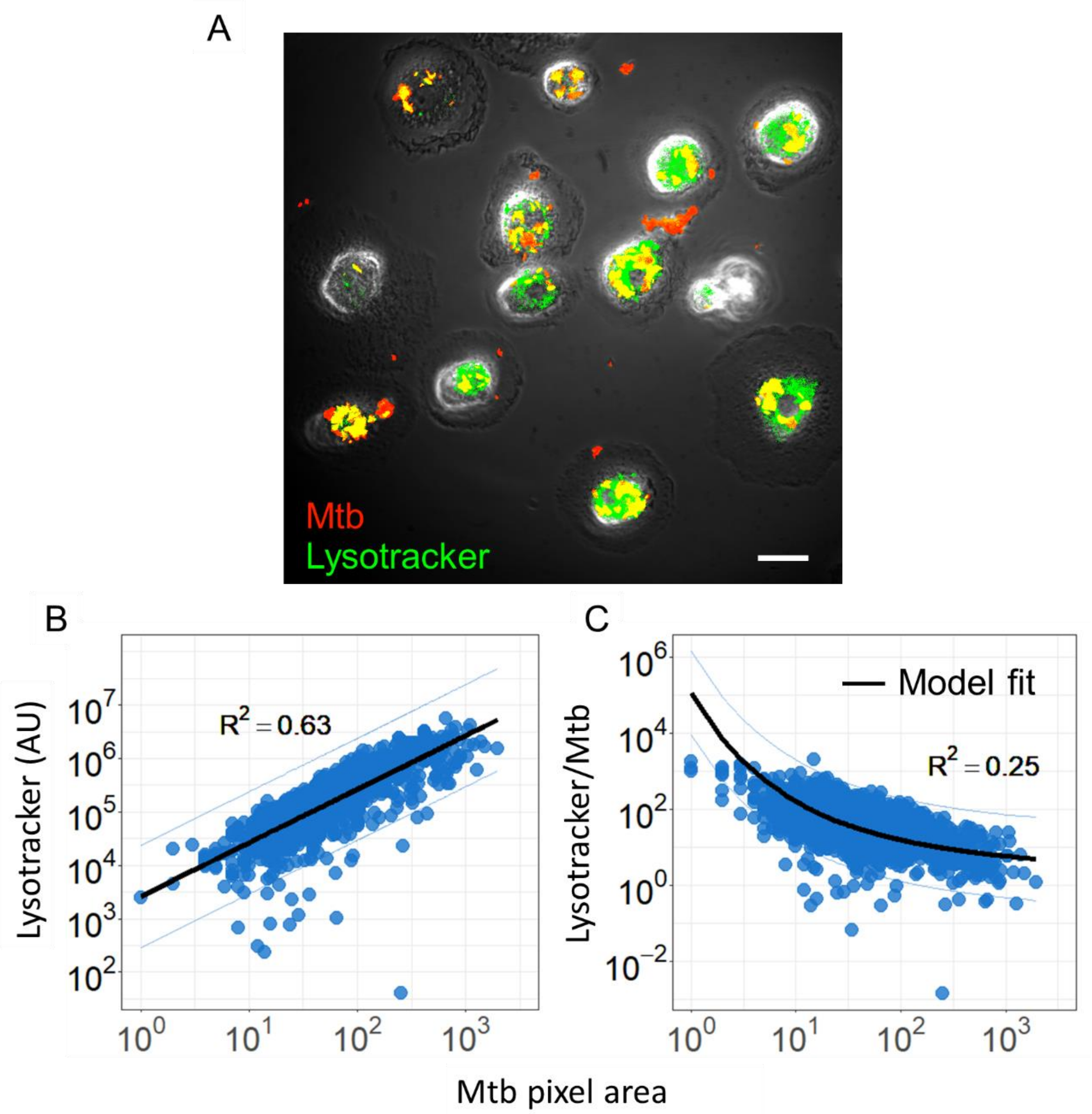


Figure 3.11 Macrophage acidification is dependent on Mtb aggregate size and is related to number of bacteria within an aggregate. Adapted w/o permission from Rodel *et al.* 2021 [162]

Image of lysotracker (green) colocalization with phagocytosed mCherry expressing Mtb (red) (A). Scale bar is 20μm. Lysotracker fluorescence as a function of total aggregate area (B). Blue points represent individual Mtb infected macrophages. Linear regression line is shown in black (R2= 0.63, p<0.0001). Ratio of Lysotracker fluorescence to Mtb fluorescence as a function of Mtb area (C). Black line shows a model based on a spherical surface area to volume ratio (R2= 0.25, p<0.0001, black line).

I also measured mean MDM acidification in response to infection with heat-killed Mtb aggregates. For cells infected with either heat-killed Mtb or live Mtb aggregates, I measured the ratio of total lysotracker signal intensity to the Mtb fluorescence intensity, across donors (**Figure 3.12).** Mean **a**cidification in MDMs infected with live Mtb aggregates (8.0 ± std 2.7) did not differ significantly from acidification in MDMs infected with heat-killed Mtb aggregates (11.4 ± std 6.9) (**Figure 3.12)**. Although standard deviation was notably lower in MDMs infected with live Mtb aggregates, which suggested a more regulated phagosomal acidification in macrophages containing live Mtb.

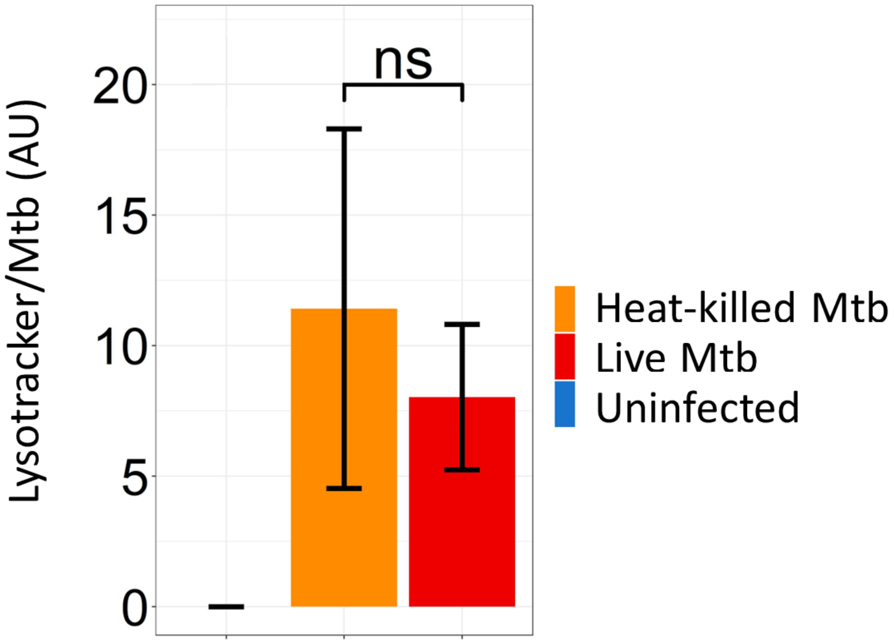


Figure 3.12 Macrophages became acidified when infected with both live and heat-killed Mtb aggregates

The ratio of lysotracker to Mtb fluorescence in MDMs infected with live Mtb (red bar), heat-killed Mtb (orange bar), or uninfected (blue bar) after 24h. Error bars indicate mean ±std and p-values were non-significant (ns) by Mann-Whitney U non-parametric test.

3.2.9 Mtb aggregates must be live to elicit death in macrophages

I investigated whether Mtb aggregates had to be alive to elicit macrophage death and whether the physical size of the internalized particles mediated the increased macrophage death rate seen when phagocytosing Mtb aggregates. I infected MDMs from 4 donors with mCherry expressing fluorescent Mtb aggregates that were either live or had been heat-killed and tracked MDM death rate (measured with membrane permeability dye) over time using time-lapse confocal fluorescence microscopy (**Figure 3.13 A - Materials and methods, section 2.3 and 2.8, Adapted from Rodel *et al.* [162]**). I saw

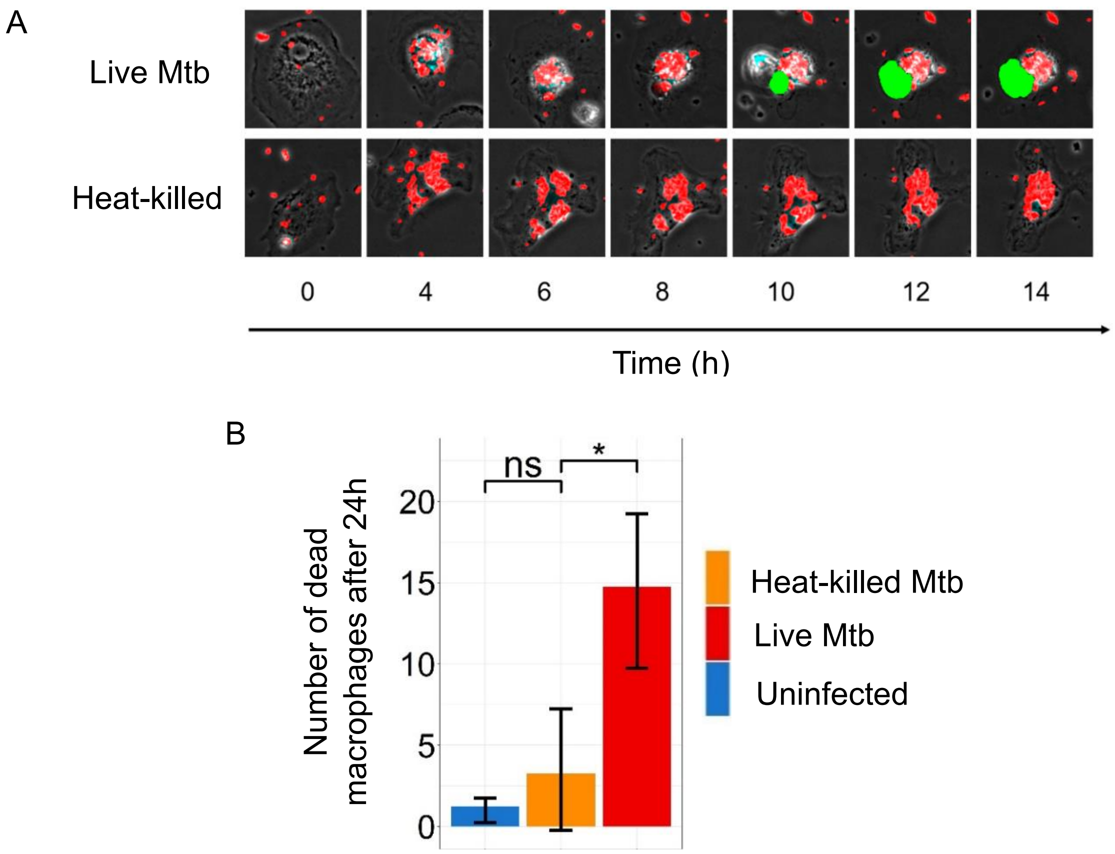


Figure 3.13: Aggregate mediated macrophage death requires live Mtb

Time-lapse microscopy showing mCherry labelled Mtb (red) induced MDM death as detected by DRAQ7 (green) (A). The number of dead cells in MDM infected with live Mtb (red bar), heat-killed Mtb (orange bar), or uninfected (blue bar) after 24h (B). Shown are mean ±std of DRAQ7 positive cells after 24h. p-value was * < 0.01 by Mann-Whitney U non-parametric test.

extensive death of MDMs infected with live Mtb aggregates. Only MDMs infected with live Mtb aggregates showed statistically significantly increased cell death after 24 hours (**Figure 12B**). Cell death in MDMs infected with heat-killed aggregates was markedly below that seen for cells infected with live Mtb and did not significantly differ from cell death levels seen for uninfected MDMs.

3.3 Chapter discussion

With this work, I aimed to identify early transcriptional signatures in macrophages that were associated with infection by Mtb aggregates. I wanted to ascertain whether transcriptional events, at early time points after infection, could characterize the enhanced macrophage death response, as demonstrated by Mahamed *et al*. in MDMs infected with Mtb aggregates and multiple single Mtb [30]. To compare between MDMs infected with multiple single Mtb bacteria and MDMs infected with Mtb aggregates that contained the same number of bacteria, I sorted infected MDMs by the fluorescence of internalized Mtb. mCherry fluorescence in this strain of H37Rv has been shown to have a tight correlation with the number of Mtb bacilli [30]. I defined a sorting gate for macrophages infected with multiple single Mtb and applied it to aggregate infected MDMs (**Figure 3.1, middle and left panel, High MOI gate).** Uninfected macrophages appeared to have a denser group of Draq7 positive cells after sorting, but this was likely due to the dispersal of Draq7 positive cell populations along the mCherry axis in Mtb infected conditions. When sorting the aggregate infected macrophages, I saw a tail of heavily infected cells extending above the sorting gate boundary that was defined in the multiple single Mtb infection (**Figure 3.1**, left panel, High MOI gate). This tail was absent in multiple single infected macrophages (**Figure 3.1**, middle panel, High MOI gate). This suggests the existence of a numerical limit to macrophage phagocytosis, which has been seen in previous studies using autologous MDMs [168]. While other studies have also identified an upper limit in terms of particle size for macrophages [169], this observation in the current study suggest that aggregated Mtb might circumvent a numerical phagocytic limit while still remaining within the phagocytic size limit. This can be visualized by considering spherical approximations of volume and surface area where volume increases faster than surface area as spherical radius increases. Higher volume, relative to physical aggregate size, translates to more space for Mtb bacilli at a relatively low cost in terms of aggregate radius. This bacterial numerical advantage could alter the macrophage response to favour the pathogen, as higher MOIs have been shown to affect the macrophage response [170, 171].

Clustering analysis had modest results when constructed using all genes in the analysis but separated along the first principal component when limiting the analysis to the most variable genes. Aggregate infected macrophages clustered farthest away from transcription profiles of uninfected macrophages, followed by multiple Mtb infected and single Mtb infected macrophages respectively (**Figure 3.2**). Genes contributing to this separation encompassed functions such as neutrophil chemotaxis and inflammation. IL-1β is one such inflammatory cytokine. Studies monitoring IL-1β levels in Mtb infected alveolar macrophages indicated that a metabolic shift to aerobic glycolysis in macrophages downregulated IL-1β and resulted in enhanced Mtb survival [172]. This suggests that IL-1β mediated inflammation is an important bacterial control mechanism. However, IL-1β has also been implicated in NLRP3 inflammasome activation and pyroptosis, a necrotic mechanism of cell death that could potentially favour the pathogen [172-174]. In the context of Mtb aggregate infection, upregulation of this cytokine could mean an increased probability of eliciting inflammatory cell death, as seen in Mahamed *et al* [30]. IL8 is a known neutrophil chemotactic factor and a potent inducer of angiogenesis [175-177]. Our transcriptional results agree with previous studies showing upregulation of IL8 in Mtb infected monocytes and in macrophages from patients with pulmonary tuberculosis [178, 179]. Mtb is also known to infect neutrophils and generating an influx of these innate responder cells could provide a niche for bacterial replication [180]. Neutrophil accumulation during Mtb infection can be detrimental to host outcomes [181, 182], and aggregated Mtb infection has been shown to lead to neutrophil accumulation in a mouse model [28]. Interestingly, angiogenesis has also been shown to be a mechanism which Mtb uses to colonize other host niches, and alongside bacterial angiogenic factors, IL8 has also been implicated in this response [183].

I used a differential expression analysis to compare transcriptional expression patterns of macrophages infected with Mtb aggregates to other infection conditions. MDMs infected with Mtb aggregates had the highest number of unique differentially regulated genes between infection conditions, relative to uninfected cells (**Figure 3.5 and 3.6**). This suggests that, even at a similar

MOI, aggregated Mtb bacilli elicit a more profound initial response to infection than Mtb phagocytosed as multiple single bacilli. This supports the idea that the enhanced death response seen in MDMs infected with aggregates is regulated by early transcriptional events [30]. I next used MAplots to visualize broad patterns of single gene expression between infection conditions (**Figure 3.6 top 3 panels**). Among highly significant differentially regulated genes (with adjusted p-value < 0.0001), aggregate infection had the highest number of upregulated genes relative to uninfected MDMs, as well as the highest number of genes with the greatest $\log_2$ fold change in expression (**Figure 3.6 top 3 panels**). Aggregate infection also had more genes upregulated, when compared directly to single Mtb infection, than multiple single Mtb compared to single Mtb. This suggests that aggregate infection elicits a unique response in infected macrophages even compared to macrophage infection with single Mtb at a similar MOI (**Figure 3.6 bottom left and middle panels**). There is also a visible cluster of genes that are highly significantly upregulated (adjusted p-value < 0.0001) with high fold change in expression, separating out above other differentially regulated genes in the aggregate versus uninfected MDM comparison (**Figure 3.6 top right panel**). This further suggests the development of a unique transcription signature that is not evident when comparing multiple single and single Mtb infection to uninfected MDMs at this early infection time point. A number of genes, including IL-1β and IL8, were commonly differentially regulated when comparing all infection conditions to uninfected macrophages.

Among the genes upregulated in all infection conditions was SERPINB2, or Plasminogen Activator Inhibitor Type 2 (PAI-2), which is involved in the regulation of the adaptive immune response [184]. It has also been found to participate in protection from TNF-α mediated cellular apoptosis [185, 186]. This suggests a mechanism by which Mtb aggregates might elicit a response that steers host cells away from apoptotic cell death and agrees with Mahamed *et al*'s observation that Mtb aggregates elicit a type of cell death that is unlike apoptosis [30]. CCL4 (Also known as the macrophage inflammatory protein MIP-1β), CCL4L1 and CCL4L2 are also commonly upregulated in infection conditions. These chemokines have been shown to have redundant function and differ primary at a single residue in the mature protein [187]. CCL4 is an inflammatory chemokine, whose secretion is partially dependent on TNF-α, and is important in the recruitment of T-helper cells during granuloma formation and the suppression intracellular growth of Mtb in alveolar macrophages [188, 189]. Superoxide dismutase 2 (SOD2) was also upregulated in all infection conditions relative to uninfected MDMs. While this gene was not present in the final refined list of differentially regulated genes (discussed below), it is interesting to note that SOD2 expression was similarly upregulated, in terms of fold change in expression, in all infection conditions (F**igure 3.6**, **Supplementary material table 8.1**). SOD2 alleviates cellular stress and prevents superoxide mediated apoptotic cell death in mouse retinal pigment epithelium [190]. However, SOD2 has also been linked to the increased apoptosis in pulmonary arterial smooth muscle cells under hypoxic conditions [191]. Interestingly, I also saw that TNFAIP6, an anti-inflammatory protein, was significantly upregulated in all infection conditions and held at a similar fold change in expression (**Figure 3.6, Supplementary material table 8.1**) [192, 193]. TNFAIP6 has also been found to promote autophagy in mouse liver cells [194]. SOD2 and TNFAIP6 therefore represent potential mechanisms through which Mtb might interact with host cell death or inflammatory pathways.

TNF and IER3 were among the genes identified when comparing aggregate to single Mtb infected MDMs (**Figure 3.6 bottom middle panels)**. TNF is known to have pluripotent effects during infection, and interestingly, this was the only gene that was significantly upregulated in the direct comparison between Multiple single and single Mtb infection of MDMs (**Figure 3.6 bottom left panel**). This suggests a role for TNF in the proportional regulation of host responses as a function of Mtb MOI. IER3, which was also identified in the direct comparison of aggregate to multiple single Mtb infection, is an inhibitor of apoptosis and may therefore be another candidate through which Mtb influences the macrophage death response [195]. HSPA1A, a heat stress protein, was also upregulated in macrophages infected with Mtb aggregates relative to multiple Mtb singles. This protein, as part of an integrated cellular stress response, has recently been found to be upregulated during necrotic murine granuloma formation in response to TNF-α stimulation, and inhibition of this response abrogates the development of necrotic granuloma and reduces Mtb bacterial burden [196]. To further characterise and define the transcriptional response of Mtb aggregates, and test for significant differences between the infection conditions, I refined the list of candidate genes by cross referencing the differential expression analysis results with the top 1% of most variable genes in the PCA analysis.

I refined our candidate gene list to a set of 21 genes that were common to both the PCA and differential expression analysis. The major trend in fold change across infection conditions was an increase in expression from uninfected to single Mtb infected, from single to multiple Mtb infected and finally peaking in aggregate infection, although this was not the case for every gene (**Figure 3.6**). IL-7R expression peaked in multiple Mtb infection and then decreased again in aggregate infection. Increased IL-7R has been associated with inflammatory disease conditions, but interestingly, a recent study has shown that the depletion of IL-7R in monocytes from patients with active tuberculosis impairs the antimycobacterial response [197]. This is especially intriguing given the IL-7R-blockade mediated upregulation of apoptosis during foetal macrophage development [198]. While the transcriptional differences for IL-7R in the current study did not test as significant**,** the trend suggests an additional means by which Mtb aggregates may be manipulating the host cell death response. Likewise, UHRF1 did not follow the predominant trend of upregulation during Mtb aggregate infection, and conversely had the lowest expression in macrophages infected with Mtb aggregates (**Figure 3.7**). The depletion of UHRF1 has been shown to induce cell cycle arrest and apoptosis, this suggests further manipulation of cell death pathways by Mtb in infected macrophages [199]. However, despite this trend, manipulation of this pathway was largely independent of MOI or aggregation state as UHRF1 expression was only significant, by nearest neighbour test for significance between infection conditions, between uninfected macrophages and single Mtb infected macrophages (**Figure 3.8**).

14 of the 21 genes in the refined list had significant differences between infection conditions. TNFα and IER3 were significantly upregulated in

aggregate infection, although the most significant differences were between uninfected and Mtb infected macrophages (of any infection type) for the set of 14 most differentially regulated genes (**Figure 3.8**). However, most genes also exhibited a trend that peaked in the aggregate infection condition (except for SERPINB2 and UHRF1). This includes TNFα and CCL4 which, along with IL-1β, have been previously identified in bronchoalveolar lavage fluid of patients with active tuberculosis [200, 201]. TNFα is well-established as playing a pivotal role in Mtb infection [202, 203]. In the absence of TNFα, mice infected with Mtb show increased bacillary load, granulomatous necrosis, susceptibility to infection and disease reactivation [202, 204, 205]. Clay *et. al* (2008) showed that an increase in mycobacterial load is connected to higher macrophage death when inhibiting TNF-α during infection and granuloma formation [206]. This highlights the active role of TNF-α in pathogen control at all stages of infection.

I investigated the collective biological functions of genes upregulated in macrophages infected with Mtb aggregates using GSEA. I found increased upregulation of genes involved in TNFα signalling and inflammation (**Figure 3.10, Table 3.1**). A study using microarrays showed a similar pattern of gene upregulation to the one identified here. Both analyses found the upregulation of pro-inflammatory mediators and TNF-α [92]. This agrees with the function of genes identified in the PCA analysis (**Section 3.2.2**). The GSEA normalized enrichment score (NES) for TNFα signalling is highest in the aggregate infection, followed by multiple single and single Mtb infection (relative to uninfected). This indicates a response by host cells to Mtb infection that is dependent on Mtb MOI and aggregation status. In addition to TNFα, inflammation was the next most highly regulated category in infection comparisons. IL-1β, also identified during PCA analysis, is an important regulator of the inflammatory response to infection [207]. Pro-IL-1β is cleaved to the mature form by caspase-1 through the inflammasome pathway, a pathway that is shared with the induction of pyroptosis [173, 208, 209]. Mahamed *et al.* noted that MDM cell death, in Mtb aggregate infected macrophages, most closely resembled a necrotic cell death induced by nigericin, a compound known for the induction of pyroptotic cell death [30, 210, 211]. This agrees with the transcriptional results seen in the current study where IER3, an inhibitor of apoptosis [195], was significantly upregulated only in aggregate infected macrophages (**Figure 3.7**). Taken together, the above suggests that TNF-α signalling, inflammation and potentially cell death manipulation are important cell level transcriptional responses of MDMs to infection with Mtb aggregates.

I also showed that acidification of macrophages infected with Mtb aggregates was related to the number of bacilli within an Mtb aggregate. The ratio of acidification to the number of Mtb bacilli decreased as the size of the aggregate increased. That is to say, the larger the Mtb aggregate, the lower the level of acidification within the macrophage per bacillus. This relationship could reasonably be described by a spherical model that approximates the SA/V ratio of an Mtb aggregate based on aggregate radius (**Figure 3.11**). This suggests that acidification may be dependent on receptor engagement during uptake within the nascent phagosome. This agrees with a large body of evidence showing that Mtb inhibits intracellular acidification to gain an advantage, even from as early as one hour post infection, and suggests that larger Mtb aggregates have an enhanced capacity to inhibit phagosome maturation relative to smaller aggregates or single Mtb [212-214]. There was no significant difference in mean acidification ratio between macrophages infected with live Mtb aggregates and heat-killed Mtb aggregates. However, acidification in heat-killed aggregate infection had notably higher variability than infection with Live Mtb aggregates (**Figure 3.12**). The absence of significant differences in acidification between heat-killed aggregates and live Mtb aggregates may be due to the heterogeneity of Mtb aggregate infections in our experiments. Mtb aggregate infection is not size controlled and contains a range of aggregates sizes, including single bacteria, that makes comparisons between acidification means challenging.

I also showed that host cell death is dependent on infection with live Mtb aggregates (**Figure 3.13**). This demonstrates that the macrophage death elicited by Mtb aggregates, shown here and by Mahamed *et al*., is not solely dependent on the physical size of the particle being phagocytosed and agrees with previous studies showing that live Mtb bacilli are required for successful pathogenesis [30, 215]. However, there is evidence showing that cellular responses are dependent on receptor engagement during phagocytosis. Such responses can be affected by the physical size, or number, of particles being phagocytosed. BC Van Der Ven *et. al* (2009) showed that macrophages primed with LPS had a higher oxidative burst when picking up reporter coated beads, relative to unstimulated macrophages, indicating that macrophage response magnitude is dependent on receptor mediated activation [216]. M. Podinovskaia *et al.* showed an effect of MOI on the macrophage response. Intensity of superoxide burst increased proportionally with the number of infecting single Mtb bacteria while proteolysis and lipolysis decreased with increasing bacterial number [171]. Using a spherical SA/V ratio, larger Mtb aggregates will contain more bacilli with proportionately lower surface area available for binding to host cell receptors, such as TLRs. An infected cell could therefore be exposed to a large amount of secreted bacterial factors that affect host cell function, such as ESAT6 or PtpA, relative to the magnitude of an antimicrobial macrophage response generated by surface receptor engagement [85, 217-219]. Said differently, the host cell may need to be "aware" of the absolute number of Mtb in an aggregate, as opposed to the number of bacteria detectable on the surface of the aggregate to mount an effective response to invading Mtb. Tying in with our anecdotal observation of a numeric phagocytic limit (**Section 3.3**), a macrophage may have a higher chance of survival if it can limit the number of bacteria it internalizes within a single phagosome. In the context of an *in vivo* infection, aggregates may therefore have an enhanced ability to overcome host cellular immune responses.

It is interesting to note that the acidification response, seen here described by a SA/V relationship related to Mtb aggregate size, inversely correlates with enhanced cell death seen in previous investigations. larger Mtb aggregates, which have previously been demonstrated to elicit increased cell death in infected MDMs, here exhibit lower acidification per bacillus in MDMs at early time points [30]. This suggests that acidification may be an important signalling event when determining macrophage cell fate at the early stages of Mtb aggregate infection.

Summarily, I found that Mtb aggregates can generate a unique transcriptional signature in infected macrophages. When compared to uninfected MDMs, aggregates significantly upregulate a higher number of genes compared to macrophages infected with multiple Mtb singles or single Mtb. In genes that were commonly regulated between these infection conditions, the trend was a peak in upregulation for aggregate infected macrophages. Genes upregulated by Mtb aggregates include those with functions in TNFα signalling and the inflammatory response. We also identified genes involved in neutrophil recruitment and regulation of the cell death response, such as IL8 and IER3 respectively. Transcriptional patterns such as these represent good candidates for further investigation of early host/pathogen interactions and thereby potentially represent novel targets for therapeutic intervention. I also found that Mtb aggregates must be alive to elicit macrophage death, and that aggregate size has an effect on MDM acidification. Interestingly, while larger Mtb aggregates generally elicit higher acidification than smaller aggregates or Mtb singles, acidification per bacillus decreases as aggregate size increases. This is important because it might indicate the mechanistic underpinnings of the Mtb aggregate manipulation of host responses.

4 Chapter 4. Development of a convolutional neural network to automate Mtb detection in tissue slides

4.1 Background

Mtb aggregates have been shown to negatively impact host response *in vitro* and in animal models [30, 148]. In order to gain an appreciation of the relevance of Mtb aggregates during a human infection, I wanted to quantify Mtb aggregates in human lung tissue. To do this, I needed to develop assistive software pipelines that would facilitate this process, now and in future investigations. Manual curation of human lung slides is a time intensive process and quantification of Mtb in a purely manual fashion would be logistically unattainable. I first developed a semi-automated digital slide analysis tool (feature extractor) that reduced the time required to analyse a stained tissue slide. I used the feature extractor, in conjunction with manual curation, to generate high quality datasets of quantified Mtb in resected human lung tissue. However, in order to expand the analysis to more tissue slides, I had to further refine the data extraction and analysis pipeline to reduce the time required for future analyses. To this end, I developed a CNN to train using our existing curated data.

Aim:

Develop a custom image analysis pipeline to automate the quantification of Ziehl-Neelson stained Mtb bacilli in histological tissue sections

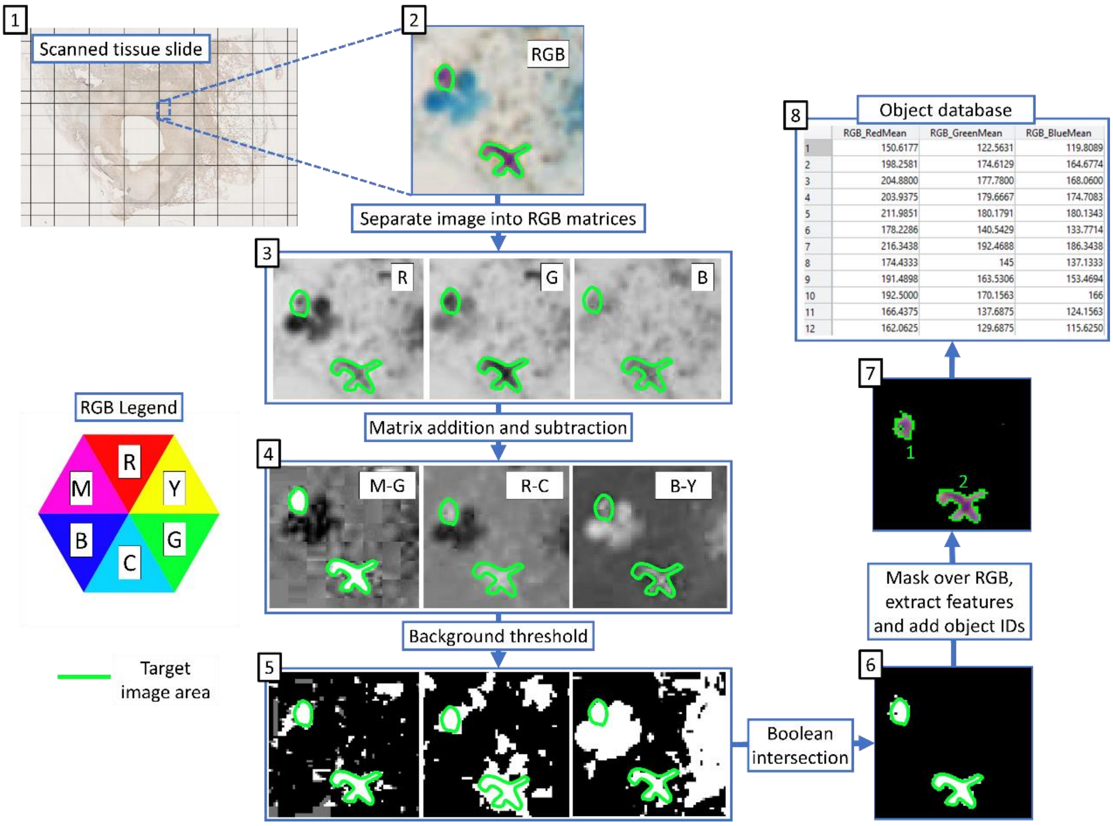
Objectives:

- Identify Ziehl Neelson stained Mtb bacilli in human lung tissue sections using a custom written MATLAB® script (Feature extractor).
- Generate a training dataset by manually validating data generated by the Feature extractor script.
- Use the newly generated training dataset to write and train a custom CNN to automate quantification of Mtb bacilli in human lung tissue sections

4.2 Results

4.2.1 ZN-stained objects can automatically be identified in tissue slides using object colour profiles

I generated datasets to train the CNN from ZN-stained resected human lung tissue. These datasets were derived from 10 scanned tissue slides from 3 Mtb positive donors who were clinically indicated for lung resection due to severe disease. The slides were randomly selected from a larger database of scanned tissue slides, with the only criterion for selection being the visible presence of ZN-Mtb. The remaining slide images in the database formed our target dataset (Chapter 5). I scanned the tissue slides into digital format prior to processing with MATLAB® image analysis scripts. Scanned slide files were too large and memory intensive to be loaded directly into MATLAB® as a single image and had to be split into smaller images (tiles) prior to processing (**Figure 4.1 – panel 1**, **materials and methods section 2.11.1, Code appendix section 8.2.1**). I identified pixels that matched the colour of ZN stained Mtb in image tiles. Each tile was imported as a full RGB image and split into individual grayscale image matrices containing red (R), green (G) or blue (B) pixel intensity information (**Figure 4.1 – panels 2 and 3**). The cyan (C), magenta (M) and yellow (Y), grayscale matrices were constructed by addition and subtraction of the R, G and B colour matrices (**Figure 4.1 - RGB Legend, materials and methods section 2.11.2).** I identified the R, G and B intensity values associated with ZN colour profile after manually identifying ZN stained Mtb in tissue sections (**section 4.2.2, Figure 4.3**). I then generated additional matrices that each uniquely biased pixel intensities towards the ZN colour profile (**Figure 4.1 – panel 4**, **materials and methods section 2.11.3**). These biased matrices each had higher pixel intensities (relative to surrounding tissue) in pixels that matched the ZN colour profile but were each uniquely derived. I then applied thresholds to the biased matrices to remove background signal (**Figure 4.1 – panel 5**, **materials and methods section 2.11.3**). I then performed a Boolean intersection operation between these three matrices. This final operation resulted in a binary mask that selected only those pixels with high intensity values in all three images (**Figure 4.1 – panel 5 and 6**). This mask was then overlaid on the original RGB input image and used to select pixels that were positive for ZN stain. The selected pixels were then grouped together by proximity so that all immediately adjacent pixels were considered part of a single object. Each of these objects was then allocated a unique identification number (**Figure 4.1 – panel 6 and 7, materials and methods section 2.11.4, Code appendix section 8.2.1**). For each of these unique objects, I then extracted 27 descriptive features. These features included object colour information, such as mean pixel intensity for each of the R, G, and B channels, or morphological features such as eccentricity or circularity (**Materials and methods section 2.11.4** for full



	RGB_RedMean	RGB_GreenMean	RGB_BlueMean
1	150.6177	122.5631	119.8089
2	198.2581	174.6129	164.6774
3	204.8800	177.7800	168.0600
4	203.9375	179.6667	174.7083
5	211.9851	180.1791	180.1343
6	178.2286	140.5429	133.7714
7	216.3438	192.4688	186.3438
8	174.4333	145	137.1333
9	191.4898	163.5306	153.4694
10	192.5000	170.1563	166
11	166.4375	137.6875	124.1563
12	162.0625	129.6875	115.6250

Figure 4.1: Schematic representation of image analysis feature extraction pipeline

Tissue slides were scanned (1-2), split into image tiles (2) and further split into separate image matrices containing R, G and B pixel intensity values (3). Bias matrices (derived by subtracting one colour matrix from another) were constructed with high intensity pixel values at positions stained with ZN (4) and thresholded to remove background signal (5). Matrices were then Boolean intersected (only positive pixels appearing in all 3 bias matrices are retained) to generate a mask (6) that was used to extract full RGB colour information (7) and generate an object database (8).

feature list). This resulted in the generation of large unvalidated datasets of ZN-stained objects (**Figure 4.1 – panel 8**). A total of 2,99 x 10^6 objects were generated this way.

4.2.2 ZN-stained Mtb bacilli must be manually identified and labelled in the datasets

I removed objects that were incorrectly identified as Ziehl Neelsen stained Mtb bacilli (ZN-Mtb) from our datasets. I did this by manually removing objects with mean RGB profiles that did not match the ZN colour profile. I plotted the mean R, G and B pixel intensities for each object on a 3-dimensional axis where the X axis was R pixel intensity, the Y axis was B pixel intensity, and the Z axis was G pixel intensity (F**igure 4.2**). I then selected those objects that fell outside

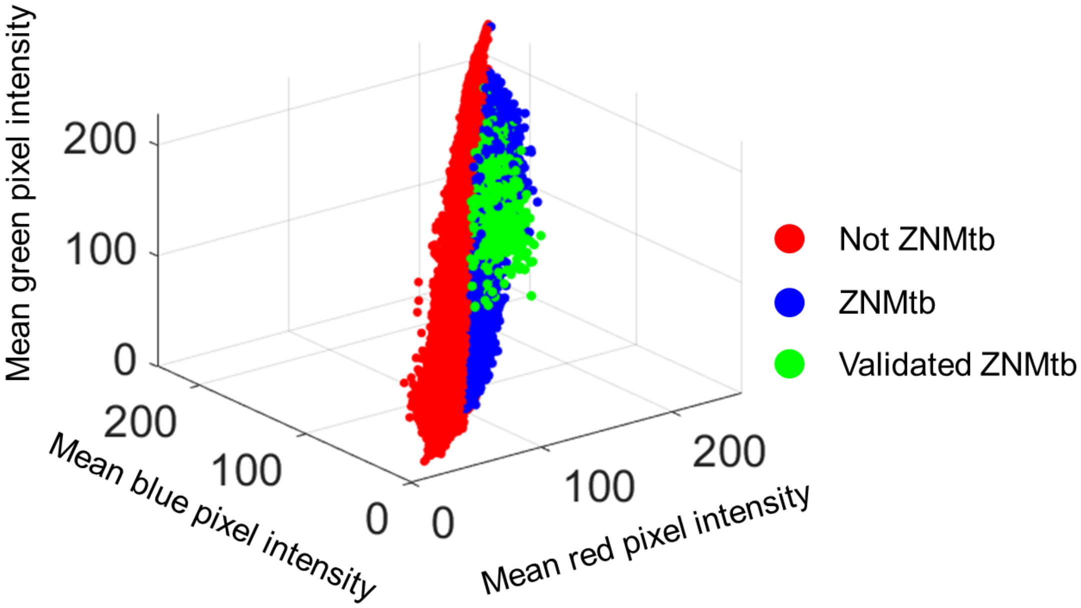


Figure 4.2: Training data for the neural network was generated by manual curation of the data

False positives (Not ZN-Mtb - red circles) were manually removed from training databases using guided manual curation. Remaining objects (ZN-Mtb) were then individually validated and assigned a true positive label if they were visually confirmed to be ZN stained Mtb bacteria (green circles).

of the ZN colour spectrum and labelled them "Not ZN-Mtb" (**Figure4.2**). This selection process was guided by plotting a separate database of objects, with colour profiles opposite to ZN-Mtb, on the same axes. This allowed us to trim the data and remove a large number of objects that were not ZN-Mtb (**Figure 4.2, materials and methods 2.11.5**, **Code appendix section 8.2.3**). I further refined the database by individually validating each of the remaining ZN-Mtb objects. I plotted the positions of ZN-Mtb objects back onto the original slide image, evaluated each object in the context of the immediate surrounding tissue, and labelled it as a true positive ZN stained Mtb object (Validated ZN-Mtb), if it was an Mtb bacillus or a group of Mtb bacilli (**Figure 4.2, materials and methods 2.11.6, Code appendix section 8.2.4**). This process consumed the most time as I validated all 2,99 x 10^6 objects, generated from each of the 10 training slides during feature extraction, in this way. Using this validated data, I also refined our estimates for the target ZN RGB colour profile I used for initial thresholding during feature extraction (**Figure 4.3**, **4.1**, **section 4.2.1**).

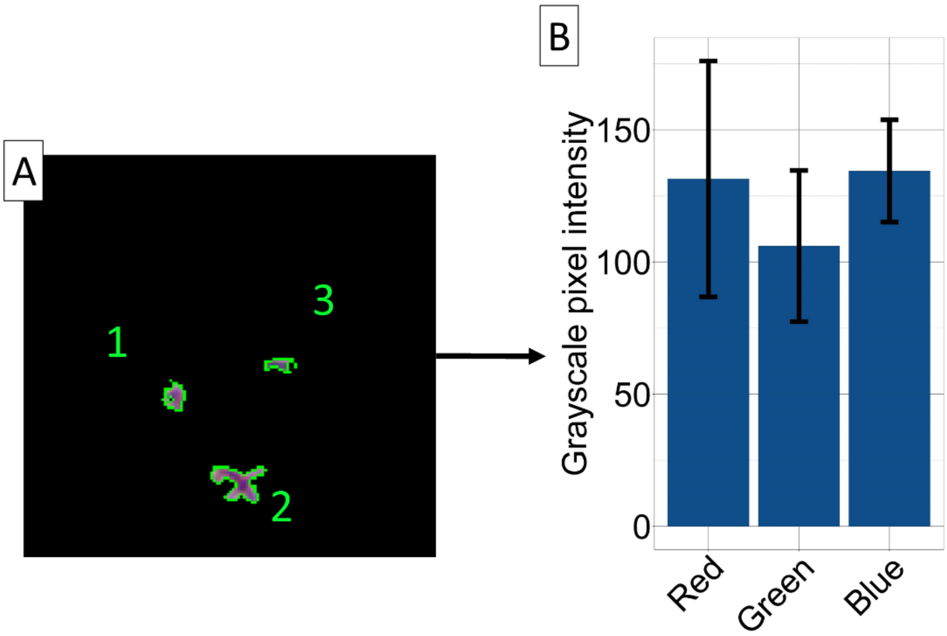


Figure 4.3: Validated Mtb objects helped define the ZN colour profile of Mtb bacilli

We used the data from manually validated Mtb objects in training datasets (A) to refine estimates for the colour profile of ZN stained Mtb used during feature extraction (B).

The feature extractor script drastically reduced the image area (measured in pixels) in images that needed to be checked for the presence of ZN-Mtb. In total, across all training set tissue slides, there was a reduction in interrogable area by a factor of 3 $\log_{10}$ orders of magnitude from 100% of the image area, down 0.37% of image area (**Figure 4.4**). The pixel area occupied by true positive ZN-Mtb however was only a fraction of this subset, at 0.004% of total pixel area (**Figure 4.4**). The resultant CNN training dataset was asymmetrical. 98.74% of the observations were labelled as "Not Mtb bacilli". A total of 37560 (1.26%) objects were labelled as ZN-Mtb. After estimating the number of bacilli (see below), I used this refined dataset as the training matrix for the CNN.

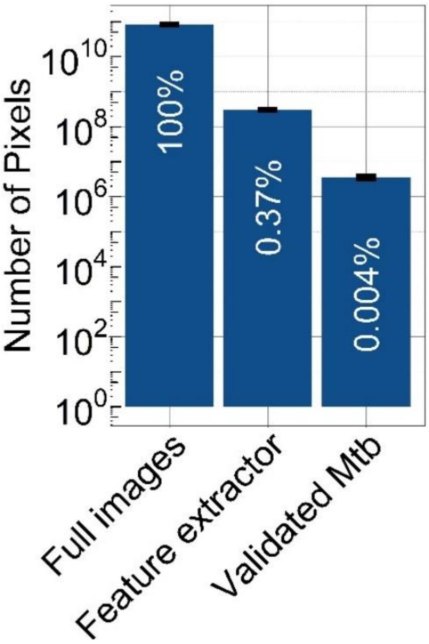


Figure 4.4: The feature extraction image analysis pipeline drastically reduces the amount of data in need of manual validation

The feature extractor eliminates greater than 99% of the image area (pixels) which need to be checked for the presence of Mtb bacilli. Given a dataset of all training images (100% left bar), the image extraction narrowed down the area where Mtb bacilli reside to 0.37% of the images (middle bar). Actual Mtb occupied 0.004% of the image (right bar), and were contained within the 0.37% of the image data identified by the feature extractor.

4.2.3 Aggregated Mtb bacilli form part of the training dataset

I estimated the total number of Mtb bacilli in our training dataset. I calculated the size of a single Mtb bacillus, in µm, after manually categorizing validated Mtb objects as either a single Mtb bacillus or multiple aggregated Mtb bacilli. I then took the average size of an Mtb bacillus and used it to estimate the number of bacteria in Mtb database objects. I used the maximum size of single bacillus as a threshold for labelling objects as ZN-Mtb aggregates (**Figure 4.5**). I extracted a total of 37560 validated ZN-Mtb objects from 10 ZN-stained lung tissue slides (**Figure 4.5**). I estimated that these 37650 objects contained approximately 66188 individual Mtb bacilli. 28017 of these bacilli were found in objects classified as ZN-Mtb aggregates (**Figure 4.5**). 42.3% of all bacilli detected in the training dataset were found as part of an Mtb aggregate.

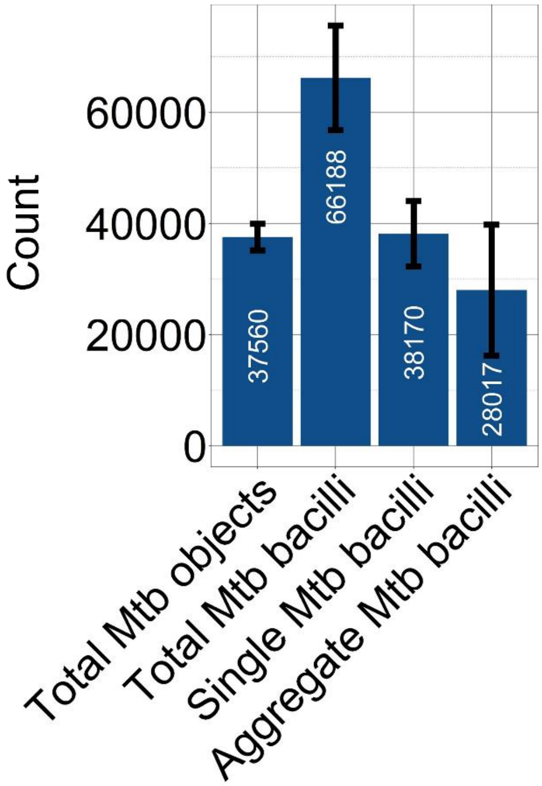


Figure 4.5: The number of Mtb objects and estimated Mtb bacilli in all training images

There were an estimated 661822 individual Mtb bacilli contained within a total of 37560 objects in the training database. 42.3% of the total estimated Mtb bacilli in the training dataset were found within Mtb aggregate objects.

4.2.4 The HyRoNet CNN classifier identifies Mtb bacilli in lung tissue sections

I wrote and trained a CNN to find ZN-Mtb in human lung tissue slides. I used the CNN to reduce the manual curation during quantification of Mtb and Mtb aggregates in large image datasets. I used a manually labelled database, (**Section 4.2.1 – 4.2.3**) as ground truths for CNN training. The first 2 neurons in the HyRoNet CNN were used to trim the dataset prior to final classification (**Figure 4.6**). A symmetrical subset of the training data (where the number of positive training examples was equal to the number of negative examples) was used to train the first neuron of the network (**Figure 4.6 Symmetrical data**

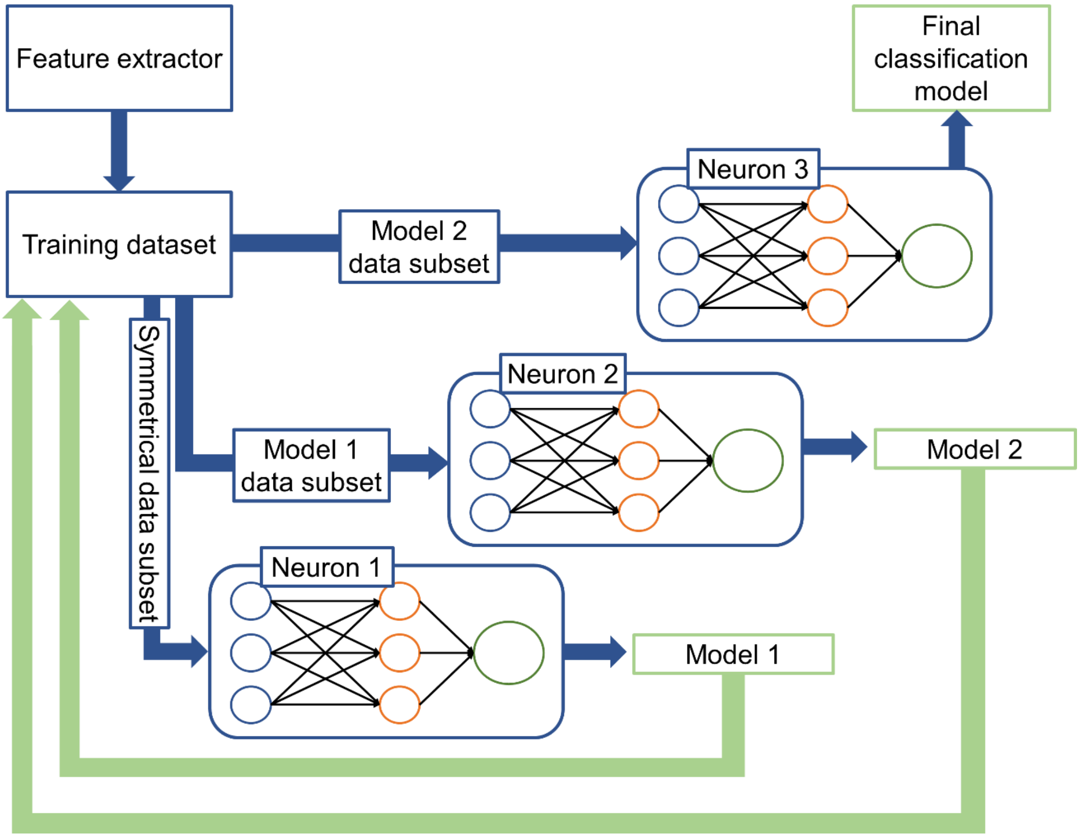


Figure 4.6: Schematic of HyRoNet training architecture

Neuron 1 of HyRoNet was used to trim asymmetrical data generated during feature extraction. Neuron 2 was used to further reduce the number of non-target observations in the dataset and facilitate model customization at neuron 3 of the network. Neuron 3 was used to calibrate final sensitivity, specificity, positive predictive (PPV) value and negative predictive value (NPV) of HyRoNet.

subset, Neuron 1, materials and methods 2.12, Code appendix section 8.2.5, 8.2.7). The resultant model classifications were then applied to the original full training dataset to generate a second data subset which was used to train neuron 2 (**Figure 4.6 Model 1 data subset, Neuron 2**). Neuron 2 generated refined classifications which were applied back to the original full training dataset to generate a third, further refined data subset (**Figure 4.6 Model 2, Model 2 data subset**). This final data subset was then used to train neuron 3, which completed the final classification model (**Figure 4.6 Neuron 3, Final classification model**). Together, these three models, and their trained parameters, constitute the HyRoNet classification algorithm.

4.2.5 Splitting of training data enables model optimization

Each neuron in HyRoNet required individual optimization to ensure reliable model training and performance. To optimise and test the performance of each neuron I performed a train, validate and test split on the training data. I randomly split the data into training and test subsets (80% training, 20% test) (**Code appendix section 8.2.6**). The test subset was used exclusively to evaluate the performance of the complete HyRoNet architecture and ensure that it generalized to unseen data. To individually optimize each neuron in the network, I further split the training dataset into a training and cross validation set (80% training, 20% cross validation). This data split technique allowed us to optimize regularization parameters and set thresholds for each neuron in the network and ensure the overall network performed well on the test data subset.

4.2.6 Individual HyRoNet neurons minimize model error

I ensured each neuron of HyRoNet achieved minimal model cost. I used forward propagation, backpropagation and gradient descent to achieve neuron convergence and the corresponding parameter matrices (**Materials and methods 2.12.2 – 2.12.4, Code appendix section 8.2.15, 8.2.16, 8.2.17**). I manually calibrated hyper parameter values by inspecting gradient descent curves, plotted as model cost per iteration of gradient descent (**Figure 4.7**).

Model convergence was indicated by a predictable reduction in model cost at each iteration of gradient descent up to horizontal asymptote (**Figure 4.7**). Unsuitable values for α, number of hidden layers and number of nodes per layer were eliminated when unpredictable gradient descent was demonstrated (**Supplementary material Figure 8.2**, **materials and methods 2.12.1 – 2.12.3, Code appendix section 8.2.5, 8.2.9**). Large values of α generated irregular cost descent curves (**Supplementary material Figure 8.2**). Adding unnecessary hidden layers to neuron architecture resulted in a delayed convergence. Layer number was therefore determined by finding values that resulted in the most rapid and predictable cost descent. Additionally, I found that feature normalization was a critical component to achieving predictable gradient descent. Training the network without normalization resulted in an erratic cost descent curve (**Supplementary material Figure 8.2, Code appendix section 8.2.8**). Neurons 1, 2 and 3 of HyRoNet achieved convergence by 5.0 x 10^4, 1.5x 10^5 and 2.0x 10^5 iterations of gradient descent respectively using a single hidden layer, each with 28 nodes (**Figure 4.7**). Values for α were 0.5 for neuron 1 and 2 and 0.4 for neuron 3.

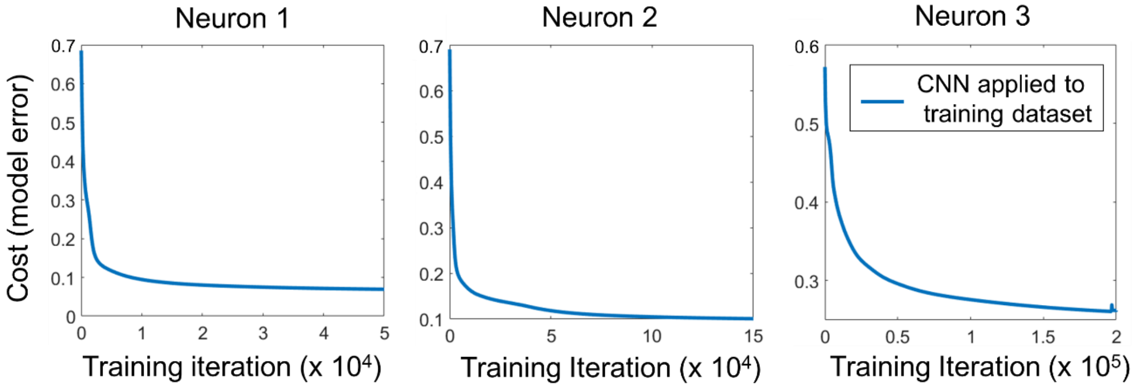


Figure 4.7: HyRoNet minimized identification errors with iterative training

Model cost (Y axis) per iteration of gradient descent (X axis) described model performance in the training dataset. Neuron 1, 2 and 3 of HyRoNet each achieved model convergence (indicated by horizontal asymptote).

4.2.7 Regularization prevents overfitting and is essential for real-world application

I prevented overfitting of the training data by automatically optimizing the regularization parameter (λ) for HyRoNet neurons. For each neuron I found the value for λ by training multiple models with different values for λ applied in each iteration (**Code appendix section 8.2.10**). I then calculated the cost for each model when applied to the training data and when applied to the cross-validation dataset and plotted these costs on the same axis as a function of λ value (**Figure 4.8**). I then found the value of λ for which the cost of the cross-validation dataset and the training dataset was lowest. I noted that only the first neuron required a regularization, as neurons 2 and 3 returned unregularized models with the lowest cost for both the training and cross validation dataset at a λ value of 0. The ideal value for λ in neuron 1 was 51.1.

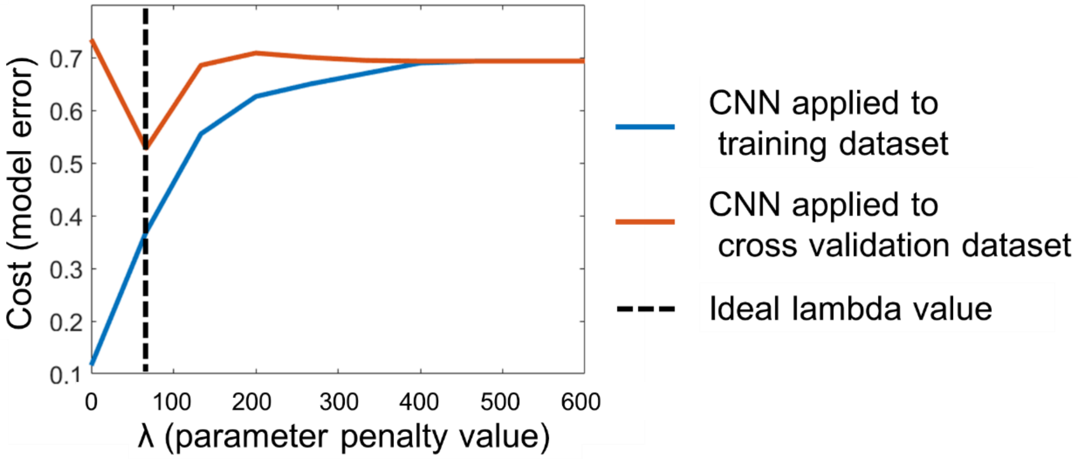


Figure 4.8: We automated regularization parameter calculation to avoid overfitting

Model cost (Y axis) when applied to a training dataset (blue line) and cross validation dataset (orange line) as a function of regularization parameter (λ). Optimal λ is indicated by the lowest attainable model cost on the cross-validation dataset. This is important as overfitting will make the CNN too specific to the training data and unsuitable for generalization to other datasets

4.2.8 HyRoNet is optimized for the removal of false positives

I customized model performance by optimizing for sensitivity in the first 2 neurons of HyRoNet and for positive predictive value (PPV) in the final neuron of the CNN (**Code appendix section 8.2.11**). I measured the area under curve (AUC) for receiver operating characteristic (ROC) curves generated by neuron 1 and 2 and found good performance at 0.9844 and 0.9931 respectively (**Figure 4.9 A, Neuron 1 and 2**). These neurons together drastically reduced the number of negatives in the training dataset by 2 $\log_{10}$ orders of magnitude from 2.7x10^6 observations in the input data to 4.2x10^4 observations in the output dataset of neuron 2. (**Figure 4.9 B, Neurons 1 and 2 output**). This allowed us to optimize neuron 3 of the network for precision and recall (sensitivity and PPV). I measured the performance of neuron 3 by PR-AUC (precision recall area under curve) and found good performance for the final neuron at 0.9735. Neuron 3 output a dataset that was composed almost entirely of ZN-Mtb observations. Note also, that the input data for neuron 3 is composed of 75% ZN-Mtb observations (**Figure 4.9 C, Neuron 2 output**), concordant with the end of the precision recall curve for neuron 3 at 75% (**Figure 4.9 A, Neuron 3**). Once data had been classified by the full HyRoNet architecture, I measured the performance of the network when I applied the model to the full training set (**Figure 4.10)**. I also measured performance on the data subset for neuron 3 only (**Supplementary material Figure 8.3**). I applied the trained network to the final test dataset and used the same metrics to measure performance. The results agreed well with those seen in the training set at an AUC for neuron 1 and 2 of 0.9853 and 0.9923 respectively and a PR-AUC 0.9735 for neuron 3. I also saw similar fold change reductions in dataset size and ratios of ZN-Mtb and similar overall model performance metrics with a final sensitivity of 80.75%, specificity of 99.96%, PPV of 96.87% and NPV of 99.7%. (**Figure 4.11 and 4.12**).

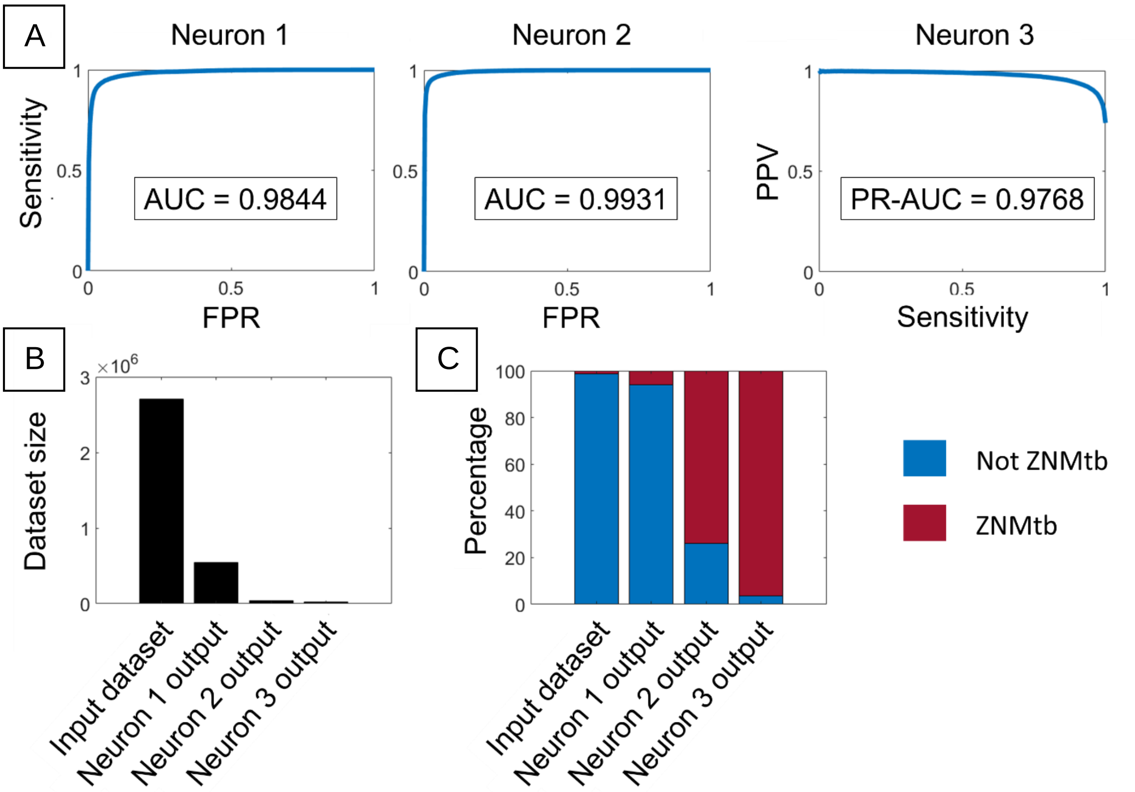


Figure 4.9: Optimizing CNN sensitivity and specificity by using multiple neurons performing different functions

Area under the curve (AUC) for Neurons 1,2 and 3 of the HyRoNet CNN applied to training data (A). Neurons 1 and 2 are optimized for sensitivity and neuron 3 is optimized for positive predictive value (PPV: number true positives/ number true positives + false positives). FPR: false positive rate. Object number decreases at each neuron (B). Progressive distillation of true Mtb objects with data flow through the neural network (C).

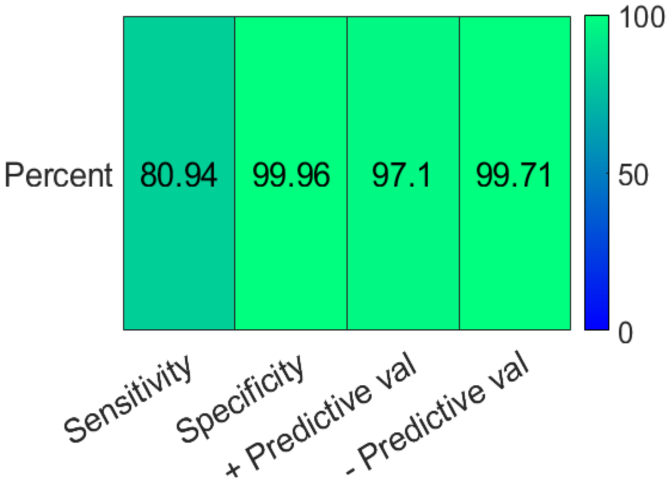


Figure 4.10: Performance estimates for training dataset

Percentage sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV, number true negatives/ number true negatives + false negatives) for HyRoNet when applied to training data. Colour key indicates percentage.

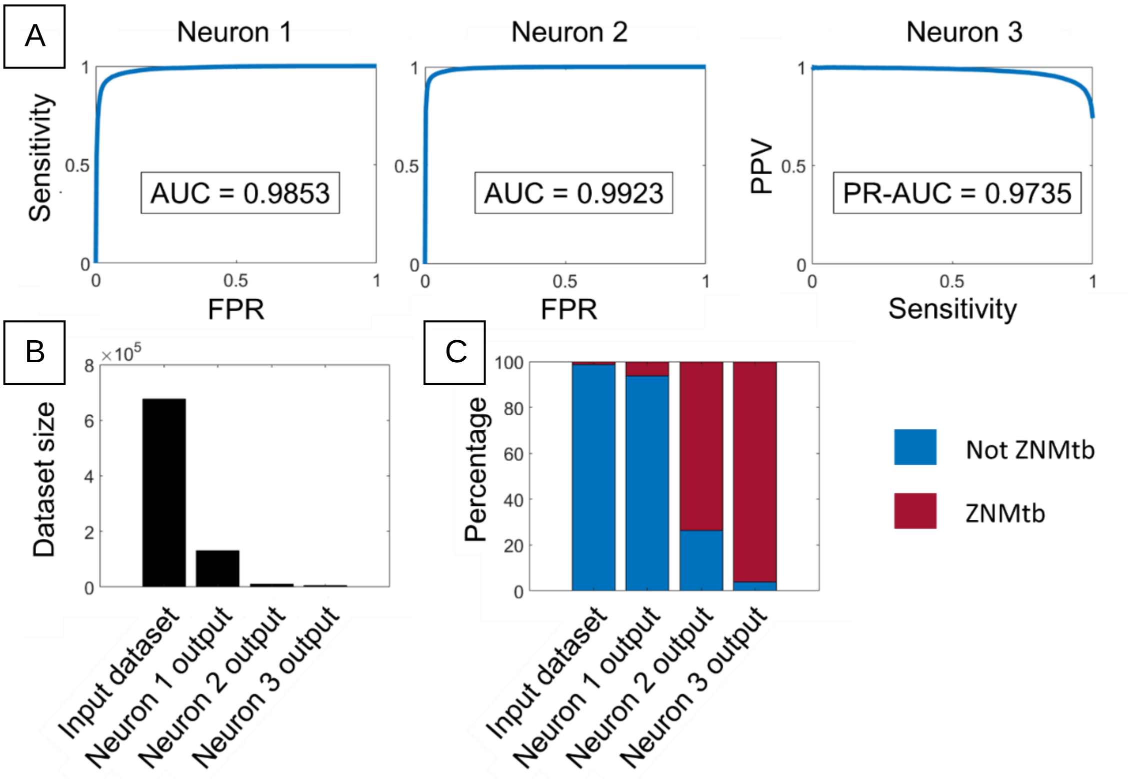


Figure 4.11: HyRoNet CNN performance on test dataset is similar to performance on training dataset

Area under the curve (AUC) for Neurons 1,2 and 3 of the HyRoNet CNN applied to test dataset (A). Neurons 1 and 2 are optimized for sensitivity and neuron 3 is optimized for positive predictive value (PPV: number true positives/ number true positives + false positives). FPR: false positive rate. Object number decreases at each neuron (B). Progressive distillation of true Mtb objects with data flow through the neural network (C).

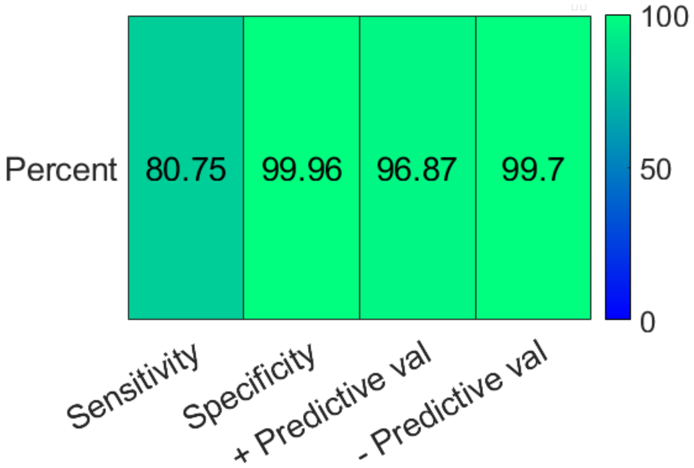


Figure 4.12: Performance estimates for test dataset

Percentage sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV, number true negatives/ number true negatives + false negatives) for HyRoNet when applied to test data. Colour key indicates percentage.

4.3 Chapter discussion

I wrote a MATLAB® script to extract pixels containing ZN stained Mtb from resected human lung tissue slide images (S**ection 4.2.1**). I used thresholds to remove background signals from each of the channels in the RGB tissue slide image. I found that, in terms of percentage area, feature extraction eliminated the largest proportion of negative pixels (pixels that were not ZN stained Mtb), leaving only 0.37% of all the pixels in our training set to be manually validated (**Figure 4.4**). However, conversion of these pixels to database objects resulted in a large database of 2,99 x 10^6 observations. Each of these objects required manual validation to ensure that they were Mtb bacilli. This was a time-consuming process and only 37560 (1.26%) of all objects were validated Mtb bacilli. The extensive manual curation required to validate these datasets highlights the need to develop tools for further automation of image analysis. Given the nature of the training dataset, containing many data points each with 27 unique descriptor variables/features, and the recent prevalence of deep learning models in similar image classification applications, a CNN was the logical model choice to create a supervised classifier [220, 221].

Manual validation of Mtb objects was an essential step in generating a reliable CNN training database. I plotted extracted Mtb objects on a 3D scatter plot and overlaid a database of objects, extracted from the same slide, composed of objects with colour profiles opposite to that of ZN-Mtb. This allowed us to select for objects that were likely to be Mtb bacilli whilst minimising subjectivity and accidental exclusion of ZN-Mtb objects. This step also allowed for the time-efficient removal of many non-target observations (objects that were Not ZN-Mtb) and thereby enabled the subsequent validation of individual ZN-Mtb objects to further refine training datasets. I manually validated each database object by visual inspection of the objects in the context of the surrounding tissue. Data validation pipelines, such as those described above, are critical steps for generating reliable training data for supervised learning models [139]. Performance of CNNs has been demonstrated to be dependent on the size and variety of a training database, but if labels are inaccurate, real world model performance will suffer regardless of database size [222]. Manual curation leverages the discrimination capacity of human-labelled data into an automated system that reduces the need for intervention during data classification [139]. In the current study, manual validation reduced interrogable pixel area percentage by two $\log_{10}$ orders of magnitude (F**igure 4.4**). This indicated that our CNN training data, while highly curated, was also asymmetrical. This would need to be accounted for when measuring CNN performance.

I wrote a custom, multi-neuron CNN to aid in automated detection of Mtb in tissue slides. I used multiple neurons because single neuron classification on asymmetrical data resulted in poor performance (**Supplementary material Figure 8.4**). The first neuron of the CNN was trained on a symmetrical subset of the data, where the number of true positives was equal to the number of true negatives, to increase overall model sensitivity [223]. Neuron 2 of the CNN was used to refine data resulting from neuron 1. This secondary data trimming step facilitated better customization of model performance, in terms of sensitivity, specificity, PPV and NPV on data fed to the final neuron of the CNN. It is noteworthy that some of the hyperparameters for the neurons in the CNN, such as number of iterations and step size of gradient descent (α), were optimized manually by monitoring cost descent curves (**Figure 4.8**). This was possible because inappropriate values generated obvious visual aberrations in cost decent curves during model training (**Supplementary material Figure 8.2)**. While automated calibration of these hyperparameters is possible, the training times of CNN models on our large datasets was prohibitive to automation [224]. However, for the regularization hyperparameter λ, where calibration directly affects model performance on unseen data, I developed an automated calculation pipeline.

I used AUC and PR-AUC to measure the performance of each neuron in the CNN. Neurons 1 and 2 of the CNN were used to trim input data and remove excess negative observations (objects that were Not ZN-Mtb) for classification by subsequent neurons. I calculated the standard AUC for ROC curves in these neurons. This allowed us to evaluate performance in terms of sensitivity and specificity so that I could calibrate these neurons to eliminate the maximum number of true negatives (FPR = 1/specificity) whilst retaining the maximum number of true positives (**Figure 4.9-4.12**). For the final neuron however, I used a PR-AUC to evaluate model performance. This was because manual validation would still have to be conducted after applying the full CNN model to real world data and using the PR-AUC as a metric allowed calibration of the model to maximize PPV. This greatly facilitated subsequent data analysis and has also been shown to be an appropriate metric to evaluate asymmetrical data [225]. Finally, it is noteworthy that accurately measuring CNN model sensitivity on real world data is challenging because it is difficult to estimate any loss (of ZN-Mtb) during feature extraction. However, I mitigated this loss by setting low background thresholds and using Boolean intersection during feature extraction.

In conclusion, I developed a semi-automated script to isolate ZN-Mtb bacilli in lung tissue sections. I manually validated data generated using this script to create a large, highly curated database to train a custom-written CNN. CNN model performance was good when evaluated on both test and training data. Together, these two programs constitute a fully automated image analysis pipeline for the quantification of Ziehl-Neelson stained Mtb bacilli in tissue sections. This can be an important tool to efficiently answer research questions surrounding Mtb pathological sites *in vivo*.

5 Chapter 5. Identifying Mtb aggregates in human lung tissue

5.1 Background

I developed image analysis pipelines to extract and quantify ZN stained Mtb, and Mtb aggregates, in human lung tissue. I first developed a feature extractor (**Section 4.2.1**) to extract data from ZN-stained tissue slides. Data resulting from feature extraction was then manually validated to ensure only ZN stained Mtb bacilli were quantified. I applied this technique to a single slide image derived from Mtb positive resected human lung tissue and analysed the resulting data (**Section 5.2.1**). I next applied the feature extractor to an additional 10 slides of Mtb positive resected human lung, that were stained for Mtb with ZN, to generate a large database of labelled observations on which to train a custom written CNN (**Section 4.2.1-4.2.4**). I trained the HyRoNet CNN using this data (**Section 4.2.5-4.2.9**). I finally wanted to apply this automated CNN to an expanded image database of 33 tissue slide images of resected human lung tissue, taken from 3 Mtb positive individuals, to quantify Mtb aggregates in *in vivo* infections.

Research questions:

- Can Mtb aggregates be automatically detected and quantified at sites of pathology in the human lung?
- Is there an association between Mtb aggregation and pathological features?
- Is there a relationship between Mtb aggregation and study participant?

Hypothesis:

Mtb aggregates can be automatically quantified at sites of pathology in the human lung

Objectives:

- Apply the newly developed HyRoNet CNN to automatically quantify Mtb bacilli in a dataset of resected human lung tissue slides

- Describe Mtb aggregate distribution at sites of pathology in resected human lung tissue slides

5.2 Results

5.2.1 Mtb aggregates are dispersed around the periphery of a granuloma cavity

I quantified Mtb in a human lung tissue slide using a custom, semi-automated image analysis pipeline. As proof of concept, I analysed a single tissue slide, from resected lung tissue of an Mtb positive donor, that had been stained for the presence of Mtb and host cell nuclei. I used our feature extractor algorithm (**Section 4.2.1)** to generate uncurated databases of ZN-Mtb and host cell nuclei objects. I validated each detected object in the context of the surrounding tissue by comparing them to a target RGB profile and classifying the object as either target or non-target observation (**Section 4.2.2-4.2.3)**. I further validated whether ZN-Mtb objects were an Mtb aggregate or an Mtb single and assigned labels accordingly. I then used this data to estimate the average size of a single Mtb bacillus and used this estimate to calculate the number of Mtb bacilli per ZN-Mtb object (**Materials and methods section 2.11.6**). I classified ZN-Mtb objects as cell associated based on whether they fell within the mean alveolar macrophage radius of host cell nuclei objects [226]. I also mapped the ZN-Mtb detections back onto the original input slide to visualize the distribution of Mtb in the tissue (**Figure 5.1, Adapted from Rodel *et al.* [162]**)

Mtb was found at the periphery of a granulomatous cavity in a tissue slide from donor PID 11302 (**Figure 5.1 A**). ZN-Mtb aggregate objects were found interspersed with ZN-Mtb objects wherever Mtb was detected. The total number of Mtb bacilli present in all objects detected in the tissue slide was estimated at 2086. 151 objects were classified as ZN-Mtb aggregates and corresponded to a minimum of 2.4 Mtb bacilli each. While the majority of ZN-Mtb objects had fewer than 2.4 bacilli (**Figure 5.1 B and C**), aggregated ZN-Mtb accounted for 28% of all detected Mtb bacilli (**Figure 5.1 C**). 993 ZN-Mtb objects, 68% of all bacilli, detected were within close proximity of host cell nuclei and were classified as cell associated. 61% of the aggregated and 70% of the single Mtb bacilli were cell associated (**Figure 5.1**).

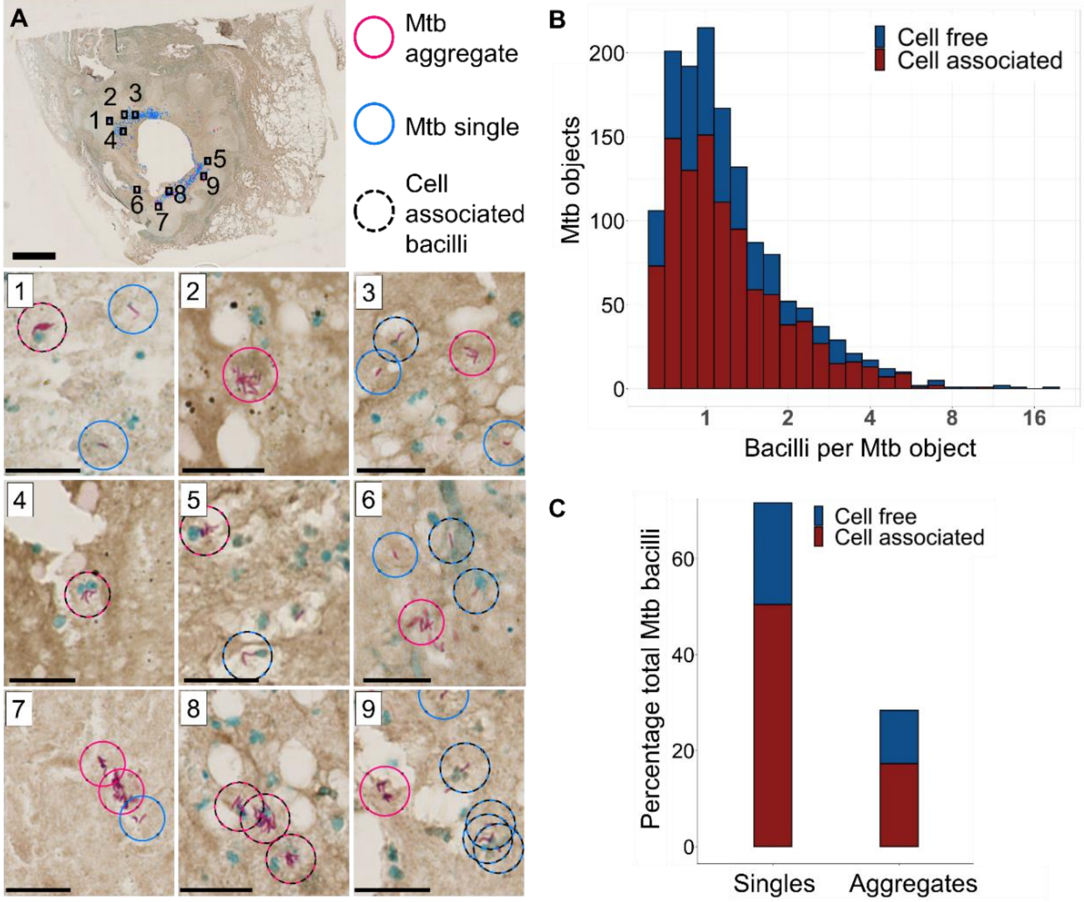


Figure 5.1: Dispersed Mtb aggregates are found near the periphery of a granuloma cavity. Adapted w/o permission from Rodel *et al.* 2021 [162]

Ziehl–Neelsen stained lung section(A). Aggregated bacilli are highlighted with a magenta circle and single bacilli with a blue circle. Scale bar is 3mm. A black perforated circle is overlaid if the Mtb are in close proximity to a cell nucleus (blue stain). Areas 1-9 are magnified in separate panels, where scale bars are 20μm (1-9). Stacked histogram of the number of cell free (blue) or cell-associated (red) Mtb objects (B). Stacked histogram of percentage of total Mtb bacilli that are single or aggregated (C).

5.2.2 HyRoNet reduces the need for manual curation, but requires a greater variety of training data

I used a trained, custom written CNN, HyRoNet, to automatically detect ZN-Mtb in human lung tissue slides. I trained HyRoNet (**Section 4.2.8**) using 10 tissue slides of resected human lung from Mtb positive individuals. Training slides had different background staining properties but were all stained for the presence of Mtb bacilli with ZN reagents. Each training slide was manually curated to generate high quality training data to facilitate the construction of a reliable CNN model (**Section 4.2.2)**. I applied the resultant trained CNN on a dataset of 33 tissue slides of resected human lung, from a total of 3 Mtb positive individuals.

I measured the overall performance of the network by calculating the percentage reduction in pixel area. Lower pixel area meant less area to manually scrutinize for the presence of ZN-Mtb. I processed each slide using the full image analysis pipeline described in **Section 4**. At each stage of the analysis pipeline I quantified the number of pixels that may contain ZN-Mtb (**Figure 5.2**). Unprocessed images contained a total of 3.1x10^{11} pixels. After feature extraction, where only those pixels that matched the ZN-Mtb RGB colour profile were retained, the total number of pixels remaining was 1.3x10^{9}. This reduction represented the greatest automated decrease in percentage pixel area and reduced the number of target pixels by 99.55%, or roughly 2 log$_{10}$ orders of magnitude (**Figure 5.2**). The feature extractor generated a database of objects/observations that was then used as input for the HyRoNet CNN for classification. Observations that were classified as “Not ZN-Mtb” were removed from the database and the total pixel area remaining was approximately 1.1x10^{8} pixels. This represented a further full log$_{10}$ magnitude reduction in percentage pixel area (**Figure 5.2**). However, manual validation was still necessary and resulted in a further reduction of 2 log$_{10}$ orders of magnitude, although this was only a small percentage of the full pixel area. This final area contained a total of 3.2x10^{5} pixels, or 0.1x10^{-4} % of the total pixel area in all input tissue slide images. I also measured the performance of HyRoNet classifications on the target dataset. Because I could only manually validate the positive predictions made by HyRoNet, I could not accurately estimate model sensitivity as I did during CNN training, where the full training dataset was manually validated. On average, HyRoNet correctly classified and removed 96.7% of true negative observations generated during feature extraction per slide (specificity of 96.7±3.8sd). However, depending on the slide being processed, there was between a 0 and 52.9% chance of an object being correctly classified as a ZN-Mtb object (PPV range 0-52.9±10.3sd).

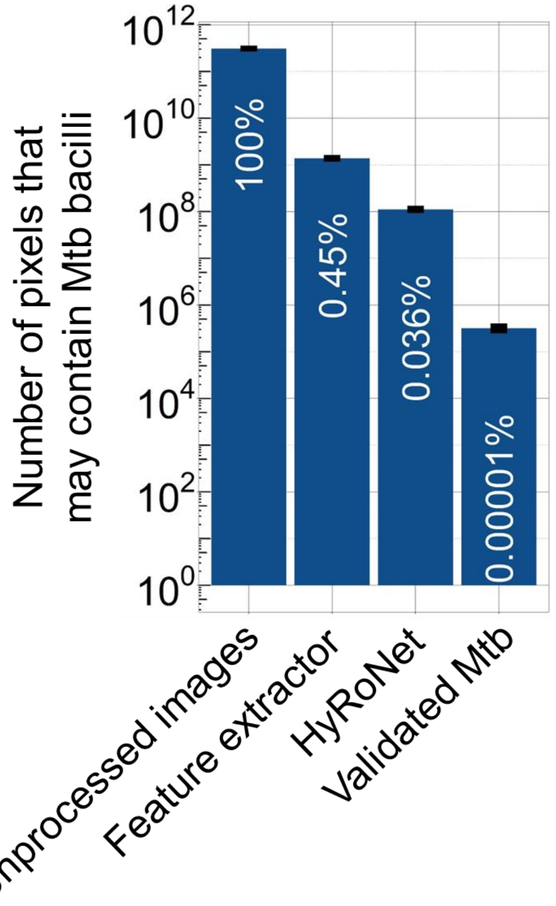


Figure 5.2: HyRoNet reduces image area in need of curation to less than 0.05% but still requires manual validation

Shown is the sum and standard deviation of pixels, that may contain Mtb, for all images at all stages of the image processing pipeline. Percentage of total dataset size indicated in white text.

5.2.3 Mtb aggregates can be found near a cavity periphery, but not exclusively

I quantified the number of Mtb aggregates, Mtb singles and the total number of Mtb bacilli in objects that were classified as ZN-Mtb by HyRoNet. Only those objects that were manually validated as ZN-Mtb were quantified (**Figure 5.3**). I calculated the number of Mtb bacilli per object, by calculating the average area for a single Mtb bacillus and using this value to estimate the number of bacilli in each object (**Materials and methods 2.11.6**). Of the 33 slides that were analysed, only 17 had detectable ZN-Mtb, and 11 slides had detectable Mtb aggregates. There were a total of 2305 single Mtb objects across all analysed slides (**Figure 5.4**). Similarly, the estimated number of single Mtb across all slides was 2650 bacilli (**Figure 5.4**). There were a total of 412 aggregate Mtb objects across all target slides. There were 3340 total estimated Mtb bacilli contained within these aggregate objects (**Figure 5.4**). The total number of Mtb bacilli contained within Mtb aggregates exceeded the number of single Mtb bacilli in the analysed tissue slides. However, the majority of these aggregated bacilli were found in a single slide (**Figure 5.6 –** PID 20314/09 slide 16). This slide contained 2851 Mtb bacilli within 283 Mtb aggregate objects. Aggregates were also found in slides 1, 3, 4 and 5 from PID 13462/09, slides 13, 14, 7, 8 and 9 from PID 21114/07 and slide 15 from PID 20314/09 (**Figure 5.5-5.6, Table 5.1 and supplementary material figure 8.5**). The remaining slides contained only single Mtb bacilli.

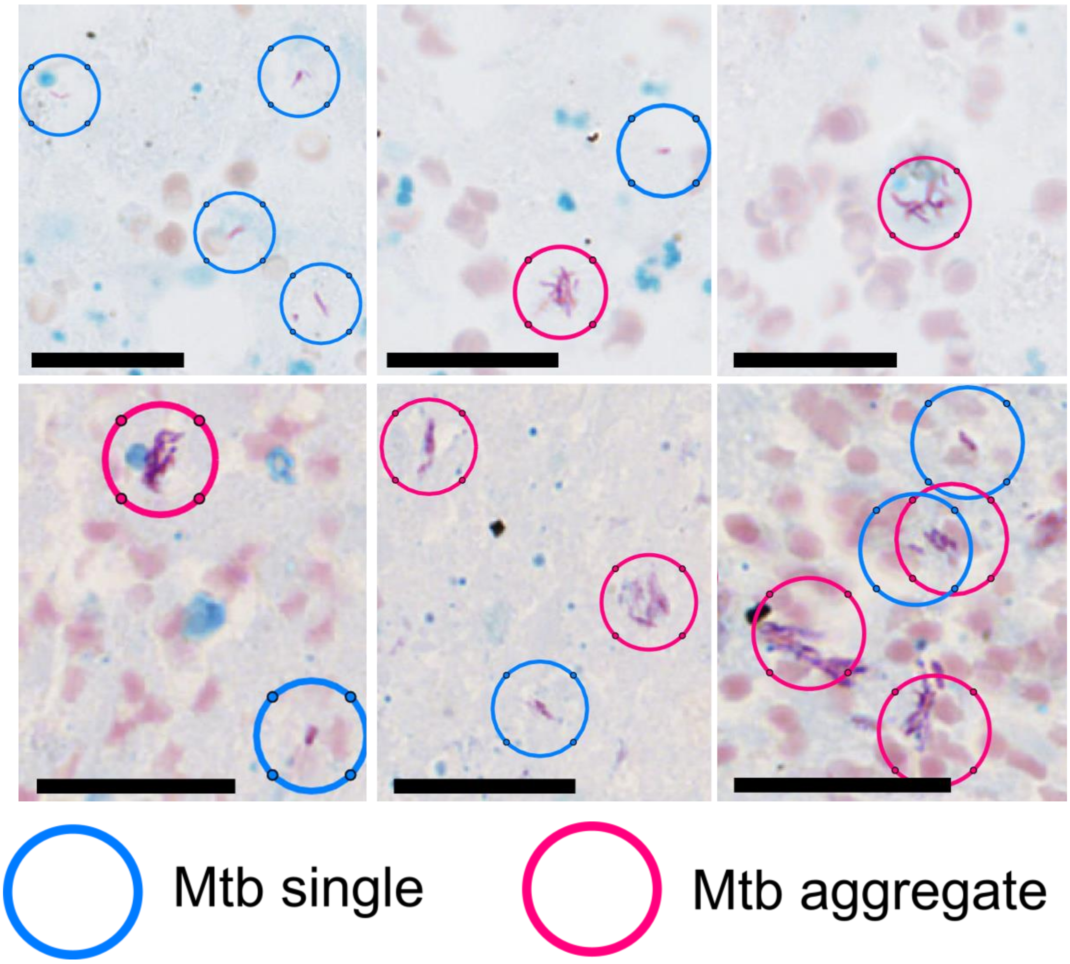


Figure 5.3: Mtb aggregates and singles can be detected in the target image database using the HyRoNet CNN

Mtb single bacteria (blue circles) and Mtb aggregates (magenta circles) are detectable in the target human lung tissue slide image database by the HyRoNet CNN. Scale bar = 25µm.

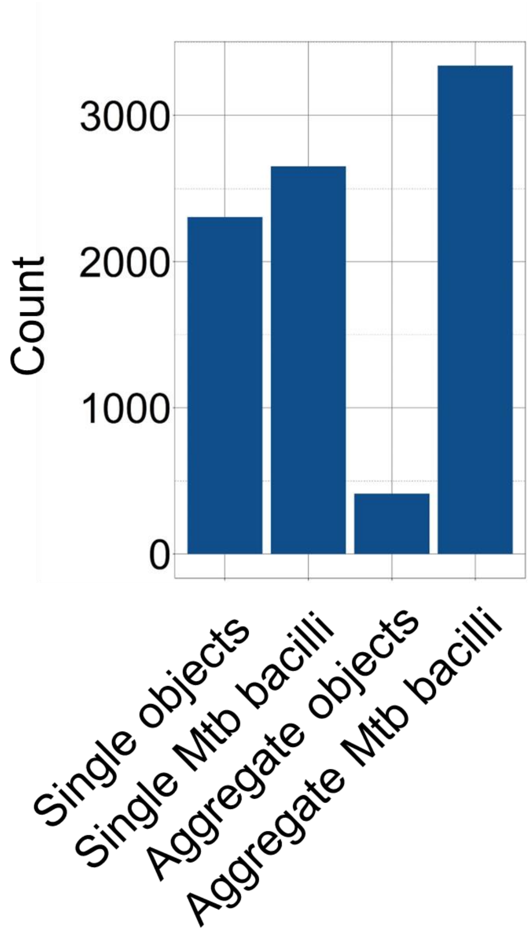


Figure 5.4: Mtb aggregate objects contain more Mtb bacilli than Mtb single objects

Objects that are classified as Mtb singles (Single objects) contain a similar number of bacilli (Single Mtb bacilli). Mtb aggregate objects (Aggregate objects) contain more individual Mtb bacilli per object (Aggregated Mtb Bacilli).

Table 5.1: Quantification of Mtb bacilli and aggregates in human lung tissue slides

PID	Tissue section	Total Mtb objects	Total Mtb bacilli	Aggregates present	Aggregate objects	Aggregate bacilli	Single objects	Single bacilli	Cavitation	Percent bacilli in aggregates
13462	3 - I ZN 0	1	4	Yes	1	4	0	0	No	100
21114	11 - P ZN 2	1	1	No	0	0	1	1	No	0
13462	17 - ZN C2	3	3	No	0	0	3	3	No	0
13462	18 - ZN C28	3	3	No	0	0	3	3	No	0
21114	9 - L ZN 9	3	7	Yes	1	4	2	3	No	57,1
21114	10 - O ZN 9	4	7	No	0	0	4	7	No	0
21114	12 - Q ZN 2	5	6	No	0	0	5	6	No	0
13462	2 - H ZN 9	39	43	No	0	0	39	43	No	0
20314	15 - ZN B	105	146	Yes	19	59	86	87	No	40,4
21114	14 - S 2	126	202	Yes	15	58	111	144	No	28,7
13462	4 - O ZN 9	166	178	Yes	3	8	163	170	No	4,5
13462	1 - F ZN 9	173	223	Yes	10	34	163	189	No	15,2
20314	16 - ZN B 2	884	3537	Yes	283	2851	601	686	No	80,6
21114	7 - E ZN 91	988	1220	Yes	43	156	945	1064	Yes	12,8

I superimposed the Mtb single and Mtb aggregate objects that I detected back onto the full scanned tissue slides. I divided the slide results into two groups. Slides in which there were fewer than 10 total Mtb bacilli detected were grouped together (**Figure 5.5**). 7 slides were included in this paucibacillary infection group. 7 slides that had more than a total of 10 detected Mtb bacilli were likewise grouped together (**Figure 5.6**). The remaining 3 re-scanned slides were included in the final bacillary count (**Supplementary material Figure 8.5**). Slides in which there were few Mtb bacilli tended to have fewer bacteria in aggregated form, although aggregates were not absent from this group. 2 of the 7 total paucibacillary slides had bacteria that were grouped into an aggregate, containing a single Mtb aggregate each (**Figure 5.5 slides 3 and 9**). In 4 of the 7 of the paucibacillary tissue slides, bacteria were found within intact granulomatous structures (**Figure 5.5 slides 3, 17, 9 and 12**). 6 of the 7 slides that contained a high number of Mtb bacilli contained Mtb aggregates (**Figure 5.6 all slides except slide 2**). In the slide in which only Mtb singles were found, bacteria were found within an intact granuloma or within a structure that resembled a lung bronchiole that was filled with acellular material (**Figure 5.6 slide 2**). For all slides in which a high number of Mtb singles were found, except for slide 7, bacteria were found within granulomatous structures (**Figure 5.6**). Slides 7 and 16 (**Figure 5.6**) contained the highest number of Mtb aggregates. Slide 7 was similar to the slide analysed using only the feature extractor (**Section 5.2.1**) where numerous aggregates were found surrounding a granulomatous cavity. This was not the case for slide 16, which had the highest number of Mtb aggregates, in which aggregates were found within an intact granuloma. Mtb singles and aggregates in this slide were found in a seam across the section (**Figure 5.6**). Mtb were also found associated with RBCs in slides 15, 16 and 2.

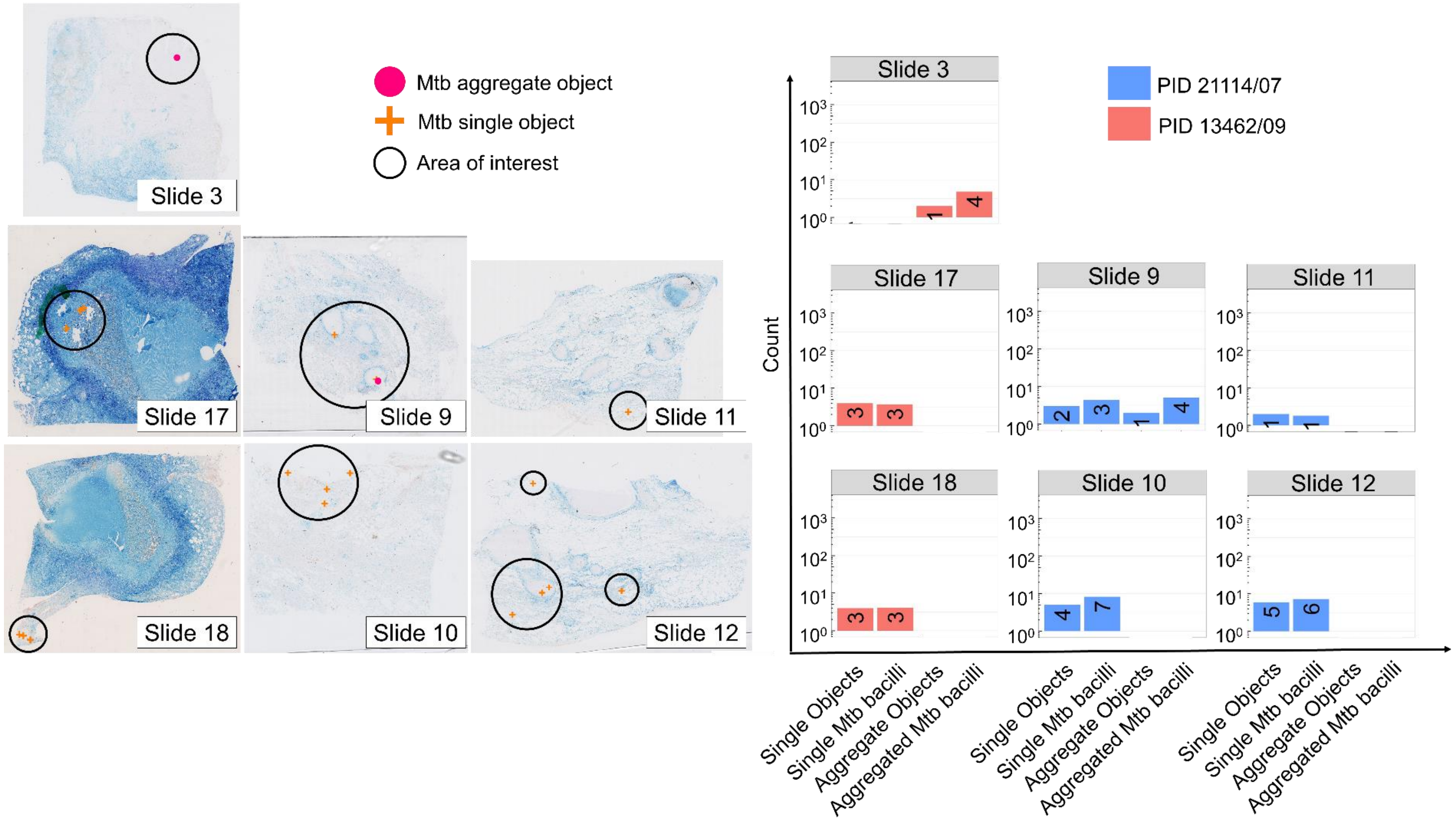


Figure 5.5: Quantification of Mtb in slides containing fewer than 10 bacilli

Tissue slides indicate locations of Mtb singles (orange crosses) and Mtb aggregates (magenta circles) within areas circled in black. Bar graphs show quantification data for Mtb singlet and aggregate objects and the estimated number of bacilli in the corresponding tissue slide. Bar colour indicates slide PID.

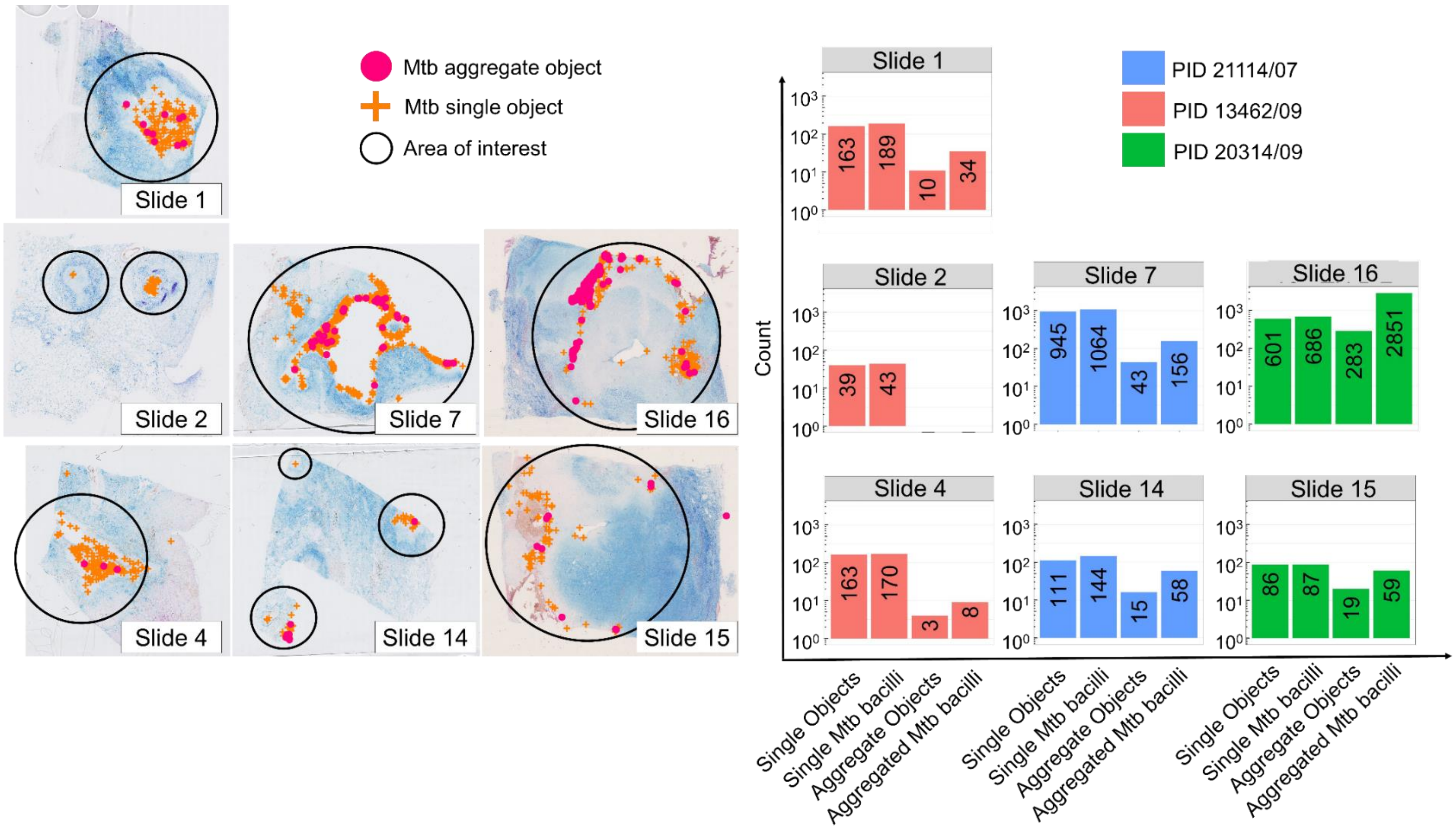


Figure 5.6: Quantification of Mtb in slides containing more than 10 bacilli

Tissue slides indicate locations of Mtb singles (orange crosses) and Mtb aggregates (magenta circles) within areas circled in black. Bar graphs show quantification data for Mtb singlet and aggregate objects and the estimated number of bacilli in the corresponding tissue slide. Bar colour indicates slide PID.

I next tested for relationships in the proportions of Mtb aggregates in tissue slides. For each slide, I plotted the number of Mtb aggregates versus the total number of Mtb objects detected per slide and fit a linear model to the data (**Figure 5.7 left panel**). I found that the number of Mtb objects had a moderate to strong positive correlation to the number of total Mtb objects per slide, and the model accounted for 38% of the variability in the data (R = 0.62, R^2 = 0.38, $p < 0.015$). I calculated the percentage of Mtb objects that were Mtb aggregates per slide and plotted as a function of the total objects (**Figure 5.7 Right panel**). The linear model I fit to the data showed a weak negative correlation that did not significantly explain the variability seen in the data (R = -0.2, R^2 = 0.041, $p < 0.47$). The overall trend in this data was a constant relationship between the proportion of Mtb aggregates and the number of Mtb objects in a slide. I split the data into groups of less than or greater than 10 Mtb bacilli per slide and tested for significant differences in the proportions of Mtb aggregates, using a non-parametric U-test, and found no significant differences between these groups ($p < 0.435$). I used data for participants who had at least 3 tissue sections to check for any donor dependent effects (**Figure 5.8**). PID 13462 showed a moderate negative correlation between the proportion of Mtb aggregates and the total number of Mtb objects, although this fit to the data had a 25% probability of occurring by chance (R = -0.55, R^2 = 0.31, $p < 0.25$). The model for PID 21114 showed a mostly constant relationship in the proportion of Mtb aggregates to total Mtb objects and did not significantly explain the variability seen in the data (R = -0.038, R^2 = 0.0014, $p < 0.94$).

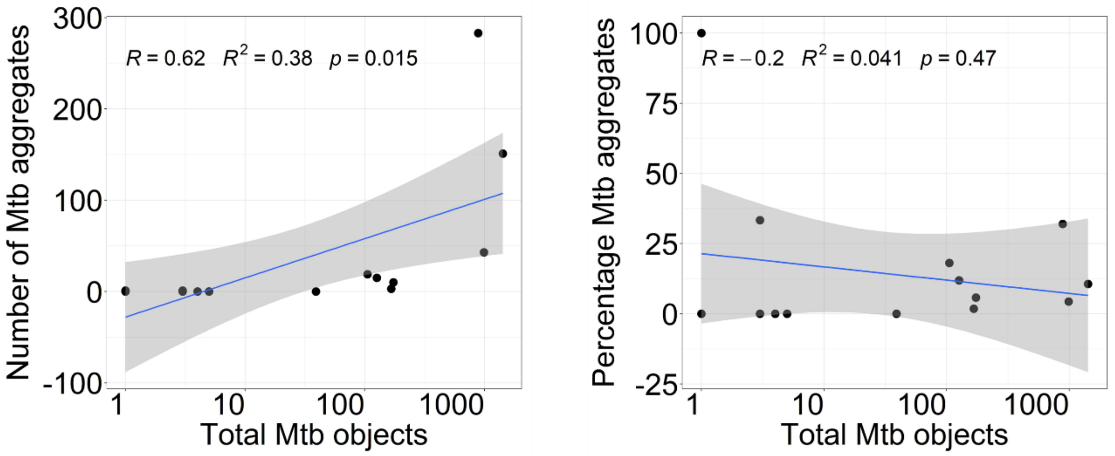


Figure 5.7: The number of Mtb aggregates per slide increases with the number of Mtb object detections per slide

Linear fit (blue line) of number (left panel) and percentage (right panel) of Mtb aggregates plotted against the total number of Mtb objects detected per slide (black circles). 95% confidence intervals shown in grey

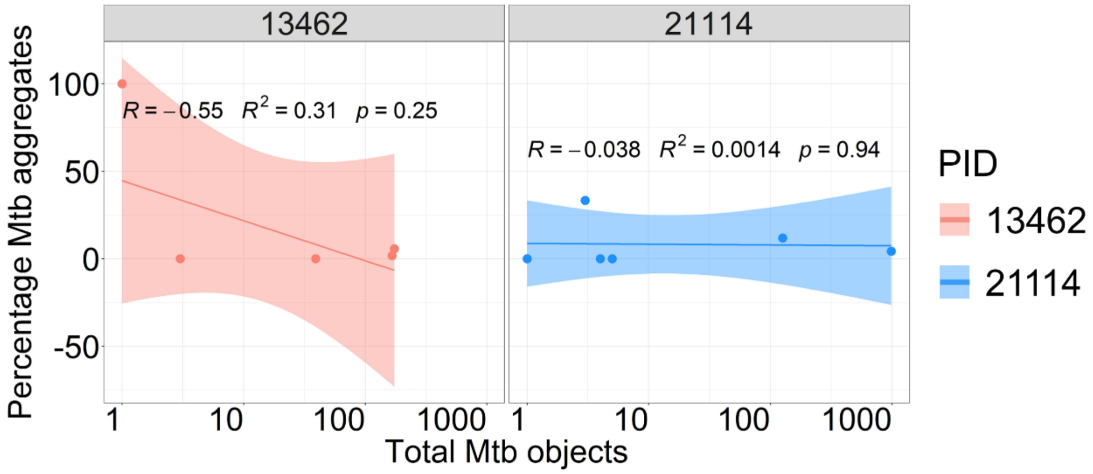


Figure 5.8: Variability in proportion of Mtb aggregates between donors

Linear fit (solid line) of percentage of Mtb aggregates plotted against the total number of Mtb objects detected per slide (coloured circles). Colour indicates participant, 95% confidence intervals shown.

5.3 Chapter discussion

I used a custom written image processing script (**Section 4.2.1**), followed by manual curation, to quantify Mtb aggregates, and cellular association, in resected human lung tissue (**Figure 5.1**). I found Mtb and Mtb aggregates at the periphery of a necrotic granuloma surrounding a cavity region. The degree of cell association between host and pathogen seen in this analysis suggests an environment where serial killing mechanics, as identified in MDMs by Mahamed *et al*., could be pertinent to host infection outcomes [30]. Additionally, transmission of Mtb aggregates has been demonstrated in rabbit aerosolized infection models and bio-aerosols [31, 148]. The physical position of Mtb aggregates surrounding a cavity, as seen in this and other studies, suggests relevance for aggregate transmission *in vivo* [147].

I used HyRoNet, a custom written CNN, trained using data generated by our feature extractor, to expand the analysis of Mtb aggregates in human lung tissue. The full HyRoNet pipeline eliminated 99.96% of pixel area from input tissue slide images. This drastically reduced the amount of data that needed to be manually curated. However, manual curation was necessary to remove false positives generated by HyRoNet predictions. It is challenging to accurately represent model performance when applied to unseen data. This is due to the heterogenous nature of stained tissue sections, which includes factors such as the actual presence of ZN-Mtb, precise tissue of origin and the chromatic variability associated with manual tissue staining. Additionally, I cannot measure the sensitivity of our model on real world data without manually validating all observations extracted by the feature extractor, at which point the analysis becomes indistinguishable from the generation of CNN training data. After manually validating HyRoNet predictions on unseen data, I found that the PPV varied more than indicated by training estimates, depending on the particular slide (**Section 5.2.2**). This was not unexpected as our training dataset was comparatively homogeneous, relative to the target dataset, and consisted of 4 tissue slides with images taken at various focal lengths (for a total of 10 training slides). While training slides provided ample individual observations for CNN training (~3.0x10^6 objects **Section 4.2.2-4.2.4**), they likely did not encompass the full range of variability associated with histological tissue sections. This was reciprocated during manual validation of CNN predictions where I found that most incorrect classifications by HyRoNet were false positive objects that resembled RBCs (**Supplementary material Figure 8.6**). This type of observation was notably absent from our training data and likely caused a drastic underestimation of PPV values in the current analysis. Therefore, by increasing the number and variety of training slides used to train HyRoNet, model performance can be further greatly improved [222]. However, HyRoNet still greatly reduced the amount of manual curation required to complete the analysis and facilitated quantification of Mtb bacilli, and Mtb aggregates, in resected human lung tissue slides.

I found Mtb aggregates in association with granuloma cavitation. Similar to findings using only the feature extractor analysis pipeline (F**igure 5.1**), HyRoNet identified a large number of Mtb aggregates surrounding a granuloma cavity (**Figure 5.6 slide 7**). Mtb aggregates have previously been identified in association with the cavity surface [147]. In tissue slides where I found fewer Mtb aggregates, but still many single Mtb bacilli, the bacteria were found in association with closed necrotic granuloma in which there was little to no evidence of cavitation (**Figure 5.6 slides 1, 4, 2 and 14**). This trend suggests that local host immunity could be better equipped to contain infections by singlet Mtb bacilli. It is however noteworthy that the slide that contained the most Mtb aggregates showed little evidence of cavitation, and instead had a dense seam of bacteria within a necrotic granuloma core (**Figure 5.6 slide 16**). These Mtb aggregates were found in close association with one another and may be the remnants of primary infection events wherein bacteria multiply within a granuloma core until the development of an adaptive response that slows infection progression [42, 43]. Conversely, slides where Mtb aggregates were found in association with cavitation appeared more disperse around the cavity surface. It may be that these dispersed Mtb aggregates develop because of exposure to a more aerobic environment that results from cavitation, and not as part of granuloma development and rupture [44]. Furthermore, the 3-dimensional structure of the granuloma was not accounted for in this analysis and would likely be informative to Mtb aggregate biology and development, as indicated by G. Wells *et al*. [124]. This is highlighted by the varying results of slides 16 and 15, which are different sections from the same block of tissue. Finally, I noted cavitation and the associated disperse Mtb aggregates, in 2 donors (PID 21114 and PID 11302), and unruptured necrotic granuloma that contained Mtb and/or Mtb aggregates, in the remaining 2 donors (PID 13462 and PID 20314). This suggests that pathogen control may be mediated by host dependent factors.

In summary, my CNN-automated image analysis pipeline was able to isolate a greatly reduced area in tissue slide images that potentially contained ZN-Mtb, and thereby greatly reduced the manual curation required to quantify Mtb bacilli. However, the CNN requires a greater training dataset to improve performance metrics. I found lung tissue slides in which Mtb aggregates were dispersed around granuloma cavities, potentially implicating Mtb aggregates in transmission-level events. I also saw a data trend that could indicate a link between participant and Mtb aggregation, but this requires a greater target dataset to validate. The current, and future further investigations on similar datasets could provide insight into the role of aggregation during the infectious lifecycle of Mtb and developing this image analysis pipeline will reduce the manual effort required for quantification of Mtb bacilli in human lung tissue.

6 Conclusions

I aimed to investigate the host transcriptional response to infection with Mtb aggregates. I showed that upregulated TNF-α expression and inflammation was a key transcriptional feature in macrophages infected with Mtb aggregates, relative to infection by multiple single Mtb at a similar MOI, or by infection with single Mtb. TNF-α is known to have pluripotent effects during cellular responses to infection and plays a role in maintaining granuloma integrity during longer term infections and in the induction of cell death [105, 106, 108]. TNF-α interacts with the NFκB signalling cascade, and increased inflammation can form part of this response [104, 227]. Counter to this, a study in 2016 showed that infection with Mtb activated STAT3 expression in macrophages, which in turn downregulated TNF and other inflammatory genes [228]. Inflammation is a key response in the recruitment of other innate and adaptive cellular effectors during Mtb infection, but this response can be co-opted to favour pathogen growth [40]. I saw that IL8, a known neutrophil chemotactic factor, was upregulated during Mtb infection. IL-1β, an inflammatory cytokine found to be upregulated in the current study, also interacts with the NFκB signalling cascade to alter cellular death responses [229]. The above highlights the complex relationship between Mtb infection, host inflammatory pathways, their effect on the cell death response, and what cell death might mean for infection trajectory. In the context of an *in vitro* Mtb aggregate infection, and the Mtb aggregate-mediated enhanced cell death seen in MDMs, the upregulated TNF-α and inflammatory response seen here appears to favour the induction of necrotic cell death that favours pathogen growth [30]. The differential regulation of other genes involved in apoptotic cell death, such as the upregulation of the apoptosis inhibitor IER3 during aggregate infection at the early time point assayed in our study, corroborates this interpretation. However, In the absence of a direct *in vivo* system of observation, I cannot directly conclude whether this sort of cell death is detrimental or beneficial for Mtb in the context of granulomatous infection trajectory.

I investigated macrophage acidification response to Mtb aggregates and found that intracellular acidification levels vary according to aggregate size, and that

Mtb aggregates need to be viable to elicit cell death in infected macrophages. Larger Mtb aggregates elicited a higher acidification response than smaller Mtb aggregates or singles in infected macrophages. However, MDMs that internalized large Mtb aggregates had lower acidification, per bacillus, than smaller Mtb aggregates or Mtb singles. The relationship between Mtb aggregates and MDM acidification, per bacillus, could be predicted by a model describing the relationship between the surface area and volume of a sphere. This suggests that acidification magnitude is dependent on receptor engagement in the nascent phagosome. This is corroborated by the fact that the cellular response to infection has previously been demonstrated to be dependent on particle size and number [171, 216]. I speculate that if signalling potential was based on this relationship, it could afford larger Mtb aggregates an advantage. A cellular response that depends on particle surface area may be lower than what might be required to control the number (volume) of bacteria in an Mtb aggregate. An infected cell could also be exposed to a higher quantity of bacterial effectors targeting the host response. Mtb aggregates may therefore circumvent the host response through a numerical advantage that is hidden from host perception.

I quantified Mtb, and Mtb aggregates, in human lung tissue sections using a custom digital image analysis pipeline that included the development of a CNN. The CNN had good performance results when evaluated using a manually validated test dataset (**Section 4.2.8**). Test dataset results reflect model performance when applied to unseen data that is similar to training data. Similar real-world performance can be achieved by using a training dataset that is more representative of the heterogeneity found in real world data to which it will be applied [230]. This was highlighted by the frequency of a specific type of false positive, resembling RBCs, found during manual validation of the CNN when applied to unseen data. This type of false positive was notably absent in the training dataset. Therefore, I can conclude that model performance will improve and better generalize to real world data through inclusion of a broader set of training examples [230]. Mtb aggregates have previously been found in association with the cavity surface [147]. Here, I found that aggregates in association with cavitous granuloma had a dispersed distribution around the cavity. I cannot say whether aggregates associated with a cavity developed before or after cavitation, perhaps as a result of a metabolic change mediated by exposure to an aerobic environment [21]. But, this does suggest a role for Mtb aggregates during transmission via their proximity to the exposed cavity surface, especially in the light of a recent studies showing aggregate transmission in patient bio-aerosols [31]. In a rabbit model of aerosolized Mtb aggregate infection, Mtb aggregates have also been shown to result in extensive necrotic foci, larger lesion size and higher bacillary load [148]. Conversely, I also found a large, closely associated group of Mtb aggregates in a necrotic granuloma that had not ruptured. Interestingly, the presence of low numbers of Mtb aggregates, even in non-cavitous granuloma, may indicate relevance for Mtb aggregates as a mechanism of pathogenicity at earlier stages of infection. I also noted that the proportion of Mtb aggregates may vary by donor PID, although more tissue sections are needed to confirm whether this trend is statistically significant. I cannot conclusively determine the relevance of aggregates during human Mtb infection through bacterial quantification alone, but their association with a failed host attempt to contain Mtb infection signals their importance.

Limitations I encountered in this study include the time taken to manually validate ZN-Mtb detected by the feature extractor to establish a training dataset. CNN model performance would benefit from a broader training dataset. However, generating reliable manually validated observations is still labour intensive. Therefore, as it stands, the current CNN model is limited in its capacity to completely automate the detection of Mtb bacilli in tissue slides without any validation. The static nature (single time-point tissue sections), and size, of the target database also limits the investigation of any dynamic roles of Mtb aggregates during the full course of an infection and current interpretations are therefore largely descriptive. Additionally, transcriptomics analysis could benefit from the inclusion of additional infection conditions, but thresholds for RNAseq cell population numbers limited further investigation.

Future work to understand the pathological significance of Mtb aggregates could include deeper investigation into the bacterial contribution during the interaction with host phagocytes. Such approaches might include a dual host/pathogen transcriptomics analysis to investigate any differences in aggregated Mtb transcription relative to singlet Mtb transcription patterns in infected macrophages. Further expansion of the automated image analysis pipeline, via the inclusion of a greater number of tissue sections, could reveal informative patterns in Mtb aggregate distributions, as well as clarify any donor dependant effects on the proportion of Mtb aggregates in infected participants. Additionally, further development of the CNN to automatically correlate any pathological tissue features with the presence of Mtb mycobacteria could provide deeper insight into tuberculous disease progression.

7 Chapter 7. References

1. Jeremiah, C., et al., *The WHO Global Tuberculosis 2021 Report–not so good news and turning the tide back to End TB.* International Journal of Infectious Diseases, 2022.
2. World Health Organization, *Global tuberculosis report 2021. License: CC BY-NC-SA 3.0 IGO.* World Health Organization, Geneva. https://www. who. int/publications/i/item/9789240037021, 2021.
3. Houben, R.M.G.J. and P.J. Dodd, *The Global Burden of Latent Tuberculosis Infection: A Re-estimation Using Mathematical Modelling.* PLOS Medicine, 2016. **13**(10): p. e1002152.
4. Koch, R., *Die aetiologie der tuberkulose.* 1882.
5. Grange, J.M., *Mycobacterium bovis infection in human beings.* Tuberculosis (Edinb), 2001. **81**(1-2): p. 71-7.
6. Sreevatsan, S., et al., *Restricted structural gene polymorphism in the Mycobacterium tuberculosis complex indicates evolutionarily recent global dissemination.* Proc Natl Acad Sci U S A, 1997. **94**(18): p. 9869-74.
7. Gutacker, M.M., et al., *Genome-wide analysis of synonymous single nucleotide polymorphisms in Mycobacterium tuberculosis complex organisms: resolution of genetic relationships among closely related microbial strains.* Genetics, 2002. **162**(4): p. 1533-43.
8. Gutierrez, M.C., et al., *Ancient origin and gene mosaicism of the progenitor of Mycobacterium tuberculosis.* PLoS Pathog, 2005. **1**(1): p. e5.
9. Novosad, S., E. Henkle, and K.L. Winthrop, *The Challenge of Pulmonary Nontuberculous Mycobacterial Infection.* Curr Pulmonol Rep, 2015. **4**(3): p. 152-161.
10. Martiniano, S.L., J.A. Nick, and C.L. Daley, *Nontuberculous Mycobacterial Infections in Cystic Fibrosis.* Clin Chest Med, 2016. **37**(1): p. 83-96.
11. Fedrizzi, T., et al., *Genomic characterization of Nontuberculous Mycobacteria.* Scientific Reports, 2017. **7**(1): p. 45258.

12. Brennan, P.J. and M.B. Goren, *Structural studies on the type-specific antigens and lipids of the mycobacterium avium. Mycobacterium intracellulare. Mycobacterium scrofulaceum serocomplex. Mycobacterium intracellulare serotype 9.* J Biol Chem, 1979. **254**(10): p. 4205-11.

13. Daffé, M., A. Quémard, and H. Marrakchi, *Biogenesis of Fatty Acids.* Lipids and Membranes, 2017: p. 1-36.

14. Chen, H., et al., *The Mycobacterial Membrane: A Novel Target Space for Anti-tubercular Drugs.* Front Microbiol, 2018. **9**: p. 1627.

15. Ortalo-Magné, A., et al., *Molecular composition of the outermost capsular material of the tubercle bacillus.* Microbiology (Reading), 1995. **141 (Pt 7)**: p. 1609-20.

16. Bansal-Mutalik, R. and H. Nikaido, *Mycobacterial outer membrane is a lipid bilayer and the inner membrane is unusually rich in diacyl phosphatidylinositol dimannosides.* Proceedings of the National Academy of Sciences, 2014. **111**(13): p. 4958-4963.

17. Chiaradia, L., et al., *Dissecting the mycobacterial cell envelope and defining the composition of the native mycomembrane.* Sci Rep, 2017. **7**(1): p. 12807.

18. Bhatt, A., et al., *Deletion of kasB in Mycobacterium tuberculosis causes loss of acid-fastness and subclinical latent tuberculosis in immunocompetent mice.* Proceedings of the National Academy of Sciences, 2007. **104**(12): p. 5157-5162.

19. Foulds, J. and R. O'brien, *New tools for the diagnosis of tuberculosis: the perspective of developing countries.* The International Journal of Tuberculosis and Lung Disease, 1998. **2**(10): p. 778-783.

20. Seiler, P., et al., *Cell-wall alterations as an attribute of Mycobacterium tuberculosis in latent infection.* The Journal of infectious diseases, 2003. **188**(9): p. 1326-1331.

21. Vilchèze, C. and L. Kremer, *Acid-fast positive and acid-fast negative Mycobacterium tuberculosis: the Koch paradox.* Microbiology spectrum, 2017. **5**(2): p. 5.2. 15.

22. Dubos, R.J. and G. Middlebrook, *The effect of wetting agents on the growth of tubercle bacilli.* J Exp Med, 1948. **88**(1): p. 81-8.

23. Leisching, G., et al., *The host response to a clinical MDR mycobacterial strain cultured in a detergent-free environment: a global transcriptomics approach.* PLoS One, 2016. **11**(4): p. e0153079.

24. Kalsum, S., et al., *The Cording Phenotype of Mycobacterium tuberculosis Induces the Formation of Extracellular Traps in Human Macrophages.* Front Cell Infect Microbiol, 2017. **7**: p. 278.

25. Ufimtseva, E.G., et al., *Mycobacterium tuberculosis cording in alveolar macrophages of patients with pulmonary tuberculosis is likely associated with increased mycobacterial virulence.* Tuberculosis, 2018. **112**: p. 1-10.

26. Middlebrook, G., R.J. Dubos, and C. Pierce, *Virulence and morphological characteristics of mammalian tubercle bacilli.* The Journal of experimental medicine, 1947. **86**(2): p. 175-184.

27. Glickman, M.S., J.S. Cox, and W.R. Jacobs, Jr., *A novel mycolic acid cyclopropane synthetase is required for cording, persistence, and virulence of Mycobacterium tuberculosis.* Mol Cell, 2000. **5**(4): p. 717-27.

28. Arias, L., et al., *Cording Mycobacterium tuberculosis bacilli have a key role in the progression towards active tuberculosis, which is stopped by previous immune response.* Microorganisms, 2020. **8**(2): p. 228.

29. Glickman, M.S., *Cording, cord factors, and trehalose dimycolate.* The mycobacterial cell envelope, 2008: p. 63-73.

30. Mahamed, D., et al., *Intracellular growth of Mycobacterium tuberculosis after macrophage cell death leads to serial killing of host cells.* eLife, 2017. **6**: p. e22028.

31. Dinkele, R., et al., *Capture and visualization of live Mycobacterium tuberculosis bacilli from tuberculosis patient bioaerosols.* PLOS Pathogens, 2021. **17**(2): p. e1009262.

32. Fennelly, K.P., et al., *Cough-generated aerosols of Mycobacterium tuberculosis: a new method to study infectiousness.* American journal of respiratory and critical care medicine, 2004. **169**(5): p. 604-609.

33. Churchyard, G., et al., *What we know about tuberculosis transmission: an overview.* The Journal of infectious diseases, 2017. **216**(suppl_6): p. S629-S635.

34. Cohen, S.B., et al., *Alveolar macrophages provide an early Mycobacterium tuberculosis niche and initiate dissemination.* Cell host & microbe, 2018. **24**(3): p. 439-446. e4.

35. Ramakrishnan, L., *Revisiting the role of the granuloma in tuberculosis.* Nat Rev Immunol, 2012. **12**(5): p. 352-66.

36. Russell, D.G., *Who puts the tubercle in tuberculosis?* Nature Reviews Microbiology, 2006. **5**: p. 39.

37. Russell, D.G., C.E. Barry, and J.L. Flynn, *Tuberculosis: What we Don't Know Can, and Does, Hurt Us.* Science, 2010. **328**(5980): p. 852.

38. Ramakrishnan, L., *Revisiting the role of the granuloma in tuberculosis.* Nature reviews. Immunology, 2012. **12**(5): p. 352.

39. Adams, D., *The granulomatous inflammatory response. A review.* The American journal of pathology, 1976. **84**(1): p. 164.

40. Volkman, H.E., et al., *Tuberculous granuloma induction via interaction of a bacterial secreted protein with host epithelium.* Science, 2010. **327**(5964): p. 466-9.

41. Marakalala, M.J., et al., *Macrophage heterogeneity in the immunopathogenesis of tuberculosis.* Frontiers in microbiology, 2018. **9**: p. 1028.

42. Swaim, L.E., et al., *Mycobacterium marinum infection of adult zebrafish causes caseating granulomatous tuberculosis and is moderated by adaptive immunity.* Infect Immun, 2006. **74**(11): p. 6108-17.

43. Jasenosky, L.D., et al., *T cells and adaptive immunity to Mycobacterium tuberculosis in humans.* Immunological reviews, 2015. **264**(1): p. 74-87.

44. Rustad, T.R., et al., *Hypoxia: a window into Mycobacterium tuberculosis latency.* Cellular microbiology, 2009. **11**(8): p. 1151-1159.

45. Barry 3rd, C.E., et al., *The spectrum of latent tuberculosis: rethinking the biology and intervention strategies.* Nature Reviews Microbiology, 2009. **7**: p. 845.

46. Lenaerts, A., C.E. Barry, 3rd, and V. Dartois, *Heterogeneity in tuberculosis pathology, microenvironments and therapeutic responses.* Immunol Rev, 2015. **264**(1): p. 288-307.

47. Cardona, P.-J., *A spotlight on liquefaction: evidence from clinical settings and experimental models in tuberculosis.* Clinical and Developmental Immunology, 2011. **2011**.

48. Flynn, J.L. and J. Chan, *Tuberculosis: latency and reactivation.* Infection and immunity, 2001. **69**(7): p. 4195-4201.

49. Ai, J.-W., et al., *Updates on the risk factors for latent tuberculosis reactivation and their managements.* Emerging microbes & infections, 2016. **5**(1): p. 1-8.

50. Lin, P.L., et al., *Sterilization of granulomas is common in active and latent tuberculosis despite within-host variability in bacterial killing.* Nature Medicine, 2013. **20**: p. 75.

51. Pisu, D., et al., *Single cell analysis of M. tuberculosis phenotype and macrophage lineages in the infected lung.* Journal of Experimental Medicine, 2021. **218**(9): p. e20210615.

52. Davies, L.C., et al., *Tissue-resident macrophages.* Nature Immunology, 2013. **14**(10): p. 986-995.

53. Schneider, C., et al., *Induction of the nuclear receptor PPAR-γ by the cytokine GM-CSF is critical for the differentiation of fetal monocytes into alveolar macrophages.* Nature immunology, 2014. **15**(11): p. 1026-1037.

54. Hashimoto, D., et al., *Tissue-resident macrophages self-maintain locally throughout adult life with minimal contribution from circulating monocytes.* Immunity, 2013. **38**(4): p. 792-804.

55. Gordon, S., *Alternative activation of macrophages.* Nature reviews immunology, 2003. **3**(1): p. 23-35.

56. Mosser, D.M. and J.P. Edwards, *Exploring the full spectrum of macrophage activation.* Nat Rev Immunol, 2008. **8**(12): p. 958-69.

57. Edwards, J.P., et al., *Biochemical and functional characterization of three activated macrophage populations.* Journal of leukocyte biology, 2006. **80**(6): p. 1298-1307.

58. Huang, L., et al., *Growth of Mycobacterium tuberculosis in vivo segregates with host macrophage metabolism and ontogeny.* Journal of Experimental Medicine, 2018. **215**(4): p. 1135-1152.

59. Cambier, C., et al., *Mycobacteria manipulate macrophage recruitment through coordinated use of membrane lipids.* Nature, 2014. **505**(7482): p. 218-222.

60. Russell, D.G., et al., *Foamy macrophages and the progression of the human tuberculosis granuloma.* Nature Immunology, 2009. **10**: p. 943.

61. Peyron, P., et al., *Foamy Macrophages from Tuberculous Patients' Granulomas Constitute a Nutrient-Rich Reservoir for M. tuberculosis Persistence.* PLOS Pathogens, 2008. **4**(11): p. e1000204.

62. Kim, M.J., et al., *Caseation of human tuberculosis granulomas correlates with elevated host lipid metabolism.* EMBO molecular medicine, 2010. **2**(7): p. 258-274.

63. Chambers, T. and W. Spector, *Inflammatory giant cells.* Immunobiology, 1982. **161**(3-4): p. 283-289.

64. Lay, G., et al., *Langhans giant cells from M. tuberculosis‐induced human granulomas cannot mediate mycobacterial uptake.* The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland, 2007. **211**(1): p. 76-85.

65. Kono, H. and K.L. Rock, *How dying cells alert the immune system to danger.* Nature Reviews Immunology, 2008. **8**(4): p. 279-289.

66. Erwig, L.-P. and P.M. Henson, *Immunological consequences of apoptotic cell phagocytosis.* The American journal of pathology, 2007. **171**(1): p. 2-8.

67. Bryant, C. and K.A. Fitzgerald, *Molecular mechanisms involved in inflammasome activation.* Trends in cell biology, 2009. **19**(9): p. 455-464.

68. Kleinnijenhuis, J., et al., *Innate immune recognition of Mycobacterium tuberculosis.* Clinical and Developmental Immunology, 2011. **2011**.

69. Underhill, D.M., et al., *Toll-like receptor-2 mediates mycobacteria-induced proinflammatory signaling in macrophages.* Proceedings of the National Academy of Sciences, 1999. **96**(25): p. 14459-14463.

70. Means, T.K., et al., *Human toll-like receptors mediate cellular activation by Mycobacterium tuberculosis.* J Immunol, 1999. **163**(7): p. 3920-7.

71. Means, T.K., et al., *Differential effects of a Toll-like receptor antagonist on Mycobacterium tuberculosis-induced macrophage responses.* The Journal of immunology, 2001. **166**(6): p. 4074-4082.

72. Sánchez, D., et al., *Role of TLR2-and TLR4-mediated signaling in Mycobacterium tuberculosis-induced macrophage death.* Cellular immunology, 2010. **260**(2): p. 128-136.

73. Ip, W.E., et al., *Phagocytosis and phagosome acidification are required for pathogen processing and MyD88-dependent responses to Staphylococcus aureus.* The Journal of Immunology, 2010. **184**(12): p. 7071-7081.

74. Queval, C.J., R. Brosch, and R. Simeone, *The macrophage: a disputed fortress in the battle against Mycobacterium tuberculosis.* Frontiers in microbiology, 2017. **8**: p. 2284.

75. Zulauf, K.E., J.T. Sullivan, and M. Braunstein, *The SecA2 pathway of Mycobacterium tuberculosis exports effectors that work in concert to arrest phagosome and autophagosome maturation.* PLoS pathogens, 2018. **14**(4): p. e1007011.

76. Chua, J., et al., *A tale of two lipids: Mycobacterium tuberculosis phagosome maturation arrest.* Current opinion in microbiology, 2004. **7**(1): p. 71-77.

77. Sun-Wada, G.-H., et al., *Direct recruitment of H+-ATPase from lysosomes for phagosomal acidification.* Journal of cell science, 2009. **122**(14): p. 2504-2513.

78. Metchnikoff, E., *Immunity in infective diseases*. 1905: University Press.

79. Vandal, O.H., et al., *A membrane protein preserves intrabacterial pH in intraphagosomal Mycobacterium tuberculosis.* Nat Med, 2008. **14**(8): p. 849-54.

80. MacMicking, J.D., *Cell-autonomous effector mechanisms against Mycobacterium tuberculosis.* Cold Spring Harbor perspectives in medicine, 2014. **4**(10): p. a018507.

81. Turner, J. and J.B. Torrelles, *Mannose-capped lipoarabinomannan in Mycobacterium tuberculosis pathogenesis.* Pathog Dis, 2018. **76**(4).

82. Fratti, R.A., et al., *Mycobacterium tuberculosis glycosylated phosphatidylinositol causes phagosome maturation arrest.* Proceedings of the National Academy of Sciences, 2003. **100**(9): p. 5437-5442.

83. Vergne, I., J. Chua, and V. Deretic, *Tuberculosis toxin blocking phagosome maturation inhibits a novel Ca2+/calmodulin-PI3K hVPS34 cascade.* The Journal of experimental medicine, 2003. **198**(4): p. 653-659.

84. Vergne, I., et al., *Mechanism of phagolysosome biogenesis block by viable Mycobacterium tuberculosis.* Proceedings of the National Academy of Sciences, 2005. **102**(11): p. 4033-4038.

85. Wong, D., et al., *Mycobacterium tuberculosis protein tyrosine phosphatase (PtpA) excludes host vacuolar-H+–ATPase to inhibit phagosome acidification.* Proceedings of the National Academy of Sciences, 2011. **108**(48): p. 19371-19376.

86. Axelrod, S., et al., *Delay of phagosome maturation by a mycobacterial lipid is reversed by nitric oxide.* Cellular Microbiology, 2008. **10**(7): p. 1530-1545.

87. Simeone, R., et al., *Cytosolic access of Mycobacterium tuberculosis: critical impact of phagosomal acidification control and demonstration of occurrence in vivo.* PLoS pathogens, 2015. **11**(2): p. e1004650.

88. Schaible, U.E., et al., *Cytokine activation leads to acidification and increases maturation of Mycobacterium avium-containing phagosomes in murine macrophages.* The Journal of Immunology, 1998. **160**(3): p. 1290-1296.

89. Ting, L.-M., et al., *Mycobacterium tuberculosis inhibits IFN-γ transcriptional responses without inhibiting activation of STAT1.* The Journal of Immunology, 1999. **163**(7): p. 3898-3906.

90. Nau, G.J., et al., *Human macrophage activation programs induced by bacterial pathogens.* Proc Natl Acad Sci U S A, 2002. **99**(3): p. 1503-8.

91. Roy, S., et al., *Transcriptional landscape of Mycobacterium tuberculosis infection in macrophages.* Sci Rep, 2018. **8**(1): p. 6758.

92. Ragno, S., et al., *Changes in gene expression in macrophages infected with Mycobacterium tuberculosis: a combined transcriptomic and proteomic approach.* Immunology, 2001. **104**(1): p. 99-108.

93. Ehrt, S., et al., *Reprogramming of the macrophage transcriptome in response to interferon-γ and Mycobacterium tuberculosis: signaling roles of nitric oxide synthase-2 and phagocyte oxidase.* The Journal of experimental medicine, 2001. **194**(8): p. 1123-1140.

94. Nathan, C.F., et al., *Identification of interferon-gamma as the lymphokine that activates human macrophage oxidative metabolism and antimicrobial activity.* The Journal of experimental medicine, 1983. **158**(3): p. 670-689.

95. Boehm, U., et al., *Cellular responses to interferon-gamma.* Annual review of immunology, 1997. **15**: p. 749.

96. Vogt, G. and C. Nathan, *In vitro differentiation of human macrophages with enhanced antimycobacterial activity.* The Journal of clinical investigation, 2011. **121**(10): p. 3889-3901.

97. Bourigault, M.L., et al., *Relative contribution of IL-1α, IL-1β and TNF to the host response to Mycobacterium tuberculosis and attenuated M. bovis BCG.* Immunity, inflammation and disease, 2013. **1**(1): p. 47-62.

98. Jayaraman, P., et al., *IL-1β promotes antimicrobial immunity in macrophages by regulating TNFR signaling and caspase-3 activation.* The Journal of immunology, 2013. **190**(8): p. 4196-4204.

99. Nathan, C. and M.U. Shiloh, *Reactive oxygen and nitrogen intermediates in the relationship between mammalian hosts and microbial pathogens.* Proceedings of the National Academy of Sciences, 2000. **97**(16): p. 8841-8848.

100. Lai, Y. and R.L. Gallo, *AMPed up immunity: how antimicrobial peptides have multiple roles in immune defense.* Trends in immunology, 2009. **30**(3): p. 131-141.

101. Watson, R.O., P.S. Manzanillo, and J.S. Cox, *Extracellular M. tuberculosis DNA targets bacteria for autophagy by activating the host DNA-sensing pathway.* Cell, 2012. **150**(4): p. 803-15.

102. Beckwith, K.S., et al., *Plasma membrane damage causes NLRP3 activation and pyroptosis during Mycobacterium tuberculosis infection.* Nature communications, 2020. **11**(1): p. 1-18.

103. Wajant, H., K. Pfizenmaier, and P. Scheurich, *Tumor necrosis factor signaling.* Cell Death & Differentiation, 2003. **10**(1): p. 45-65.

104. Liu, T., et al., *NF-κB signaling in inflammation.* Signal Transduct Target Ther, 2017. **2**: p. 17023-.

105. Roach, D.R., et al., *TNF regulates chemokine induction essential for cell recruitment, granuloma formation, and clearance of mycobacterial infection.* The Journal of immunology, 2002. **168**(9): p. 4620-4627.

106. Kindler, V., et al., *The inducing role of tumor necrosis factor in the development of bactericidal granulomas during BCG infection.* Cell, 1989. **56**(5): p. 731-740.

107. Keane, J., et al., *Tuberculosis associated with infliximab, a tumor necrosis factor α–neutralizing agent.* New England Journal of Medicine, 2001. **345**(15): p. 1098-1104.

108. Roca, F.J., et al., *TNF induces pathogenic programmed macrophage necrosis in tuberculosis through a mitochondrial-lysosomal-endoplasmic reticulum circuit.* Cell, 2019. **178**(6): p. 1344-1361. e11.

109. Balcewicz-Sablinska, M.K., et al., *Pathogenic Mycobacterium tuberculosis evades apoptosis of host macrophages by release of TNF-R2, resulting in inactivation of TNF-α.* The journal of Immunology, 1998. **161**(5): p. 2636-2641.

110. Chen, M., H. Gan, and H.G. Remold, *A mechanism of virulence: virulent Mycobacterium tuberculosis strain H37Rv, but not attenuated H37Ra, causes significant mitochondrial inner membrane disruption in macrophages leading to necrosis.* The Journal of Immunology, 2006. **176**(6): p. 3707-3716.

111. Keane, J., H.G. Remold, and H. Kornfeld, *Virulent Mycobacterium tuberculosis strains evade apoptosis of infected alveolar macrophages.* J Immunol, 2000. **164**(4): p. 2016-20.

112. Lee, J., M. Hartman, and H. Kornfeld, *Macrophage apoptosis in tuberculosis.* Yonsei medical journal, 2009. **50**(1): p. 1-11.

113. Lam, A., et al., *Role of apoptosis and autophagy in tuberculosis.* Am J Physiol Lung Cell Mol Physiol, 2017. **313**(2): p. L218-l229.

114. Ramon-Luing, L.A., et al., *Diverse Cell Death Mechanisms Are Simultaneously Activated in Macrophages Infected by Virulent Mycobacterium tuberculosis.* Pathogens, 2022. **11**(5): p. 492.

115. Lee, J., et al., *Macrophage apoptosis in response to high intracellular burden of Mycobacterium tuberculosis is mediated by a novel caspase-independent pathway.* Journal of immunology (Baltimore, Md. : 1950), 2006. **176**: p. 4267-4274.

116. Davis, J.M. and L. Ramakrishnan, *The role of the granuloma in expansion and dissemination of early tuberculous infection.* Cell, 2009. **136**(1): p. 37-49.

117. Repasy, T., et al., *Intracellular bacillary burden reflects a burst size for Mycobacterium tuberculosis in vivo.* PLoS Pathog, 2013. **9**(2): p. e1003190.

118. Fisher, M.A., B.B. Plikaytis, and T.M. Shinnick, *Microarray analysis of the Mycobacterium tuberculosis transcriptional response to the acidic conditions found in phagosomes.* Journal of bacteriology, 2002. **184**(14): p. 4025-4032.

119. Zhao, S., et al., *Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells.* PloS one, 2014. **9**(1): p. e78644.

120. Thawng, C.N. and G.B. Smith, *A transcriptome software comparison for the analyses of treatments expected to give subtle gene expression responses.* BMC genomics, 2022. **23**(1): p. 1-12.

121. Li, D., et al., *An evaluation of RNA-seq differential analysis methods.* bioRxiv, 2022.

122. Love, M.I., W. Huber, and S. Anders, *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.* Genome biology, 2014. **15**(12): p. 1-21.

123. Ulrichs, T. and S.H. Kaufmann, *New insights into the function of granulomas in human tuberculosis.* The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland, 2006. **208**(2): p. 261-269.

124. Wells, G., et al., *μCT Analysis of the Human Tuberculous Lung Reveals Remarkable Heterogeneity in 3D Granuloma Morphology.* American Journal of Respiratory and Critical Care Medicine, 2021.

125. Bankhead, P., et al., *QuPath: Open source software for digital pathology image analysis.* Scientific reports, 2017. **7**(1): p. 1-7.

126. Collins, T.J., *ImageJ for microscopy.* Biotechniques, 2007. **43**(S1): p. S25-S30.

127. Zhang, J., et al., *A comprehensive review of image analysis methods for microorganism counting: from classical image processing to deep learning approaches.* Artificial Intelligence Review, 2022. **55**(4): p. 2875-2944.

128. Payasi, Y. and S. Patidar. *Diagnosis and counting of tuberculosis bacilli using digital image processing*. in *2017 international conference on information, communication, instrumentation and control (ICICIC)*. 2017. IEEE.

129. Shen, D., G. Wu, and H.-I. Suk, *Deep learning in medical image analysis.* Annual review of biomedical engineering, 2017. **19**: p. 221-248.

130. Komura, D. and S. Ishikawa, *Machine learning methods for histopathological image analysis.* Computational and structural biotechnology journal, 2018. **16**: p. 34-42.

131. Yamashita, R., et al., *Convolutional neural networks: an overview and application in radiology.* Insights into Imaging, 2018. **9**(4): p. 611-629.

132. Alzubaidi, L., et al., *Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions.* Journal of big Data, 2021. **8**(1): p. 1-74.

133. Baştanlar, Y. and M. Özuysal, *Introduction to machine learning.* miRNomics: MicroRNA biology and computational analysis, 2014: p. 105-128.

134. Burkov, A., *The hundred-page machine learning book*. Vol. 1. 2019: Andriy Burkov Quebec City, QC, Canada.

135. Ruder, S., *An overview of gradient descent optimization algorithms.* arXiv preprint arXiv:1609.04747, 2016.

136. Tian, Y. and Y. Zhang, *A comprehensive survey on regularization strategies in machine learning.* Information Fusion, 2022. **80**: p. 146-166.

137. Sathya, R. and A. Abraham, *Comparison of supervised and unsupervised learning algorithms for pattern classification.* International Journal of Advanced Research in Artificial Intelligence, 2013. **2**(2): p. 34-38.

138. Tarca, A.L., et al., *Machine learning and its applications to biology.* PLoS computational biology, 2007. **3**(6): p. e116.

139. Kotsiantis, S.B., I. Zaharakis, and P. Pintelas, *Supervised machine learning: A review of classification techniques.* Emerging artificial intelligence applications in computer engineering, 2007. **160**(1): p. 3-24.

140. Ghahramani, Z. *Unsupervised learning*. in *Summer school on machine learning*. 2003. Springer.

141. Hosmer Jr, D.W., S. Lemeshow, and R.X. Sturdivant, *Applied logistic regression*. Vol. 398. 2013: John Wiley & Sons.

142. Dreiseitl, S. and L. Ohno-Machado, *Logistic regression and artificial neural network classification models: a methodology review.* Journal of biomedical informatics, 2002. **35**(5-6): p. 352-359.

143. Wang, S.-C., *Artificial neural network*, in *Interdisciplinary computing in java programming*. 2003, Springer. p. 81-100.

144. Albawi, S., T.A. Mohammed, and S. Al-Zawi. *Understanding of a convolutional neural network*. in *2017 international conference on engineering and technology (ICET)*. 2017. Ieee.

145. Widrow, B. and M.A. Lehr, *30 years of adaptive neural networks: perceptron, madaline, and backpropagation.* Proceedings of the IEEE, 1990. **78**(9): p. 1415-1442.

146. Werbos, P., *Beyond regression:" new tools for prediction and analysis in the behavioral sciences.* Ph. D. dissertation, Harvard University, 1974.

147. Kaplan, G., et al., *Mycobacterium tuberculosis growth at the cavity surface: a microenvironment with failed immunity.* Infect Immun, 2003. **71**(12): p. 7099-108.

148. Kolloli, A., et al., *Aggregation state of Mycobacterium tuberculosis impacts host immunity and augments pulmonary disease pathology.* Communications biology, 2021. **4**(1): p. 1-12.

149. Díaz-Huerta, J.L., et al., *Image processing for AFB segmentation in bacilloscopies of pulmonary tuberculosis diagnosis.* Plos one, 2019. **14**(7): p. e0218861.

150. Shah, M.I., et al., *Ziehl–Neelsen sputum smear microscopy image database: a resource to facilitate automated bacilli detection for tuberculosis diagnosis.* Journal of Medical Imaging, 2017. **4**(2): p. 027503.

151. Panicker, R.O., et al., *Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods.* Biocybernetics and Biomedical Engineering, 2018. **38**(3): p. 691-699.

152. Sadaphal, P., et al., *Image processing techniques for identifying Mycobacterium tuberculosis in ZN stains.* The International Journal of Tuberculosis and Lung Disease, 2008. **12**(5): p. 579-582.

153. Trombetta, J.J., et al., *Preparation of single‐cell RNA‐seq libraries for next generation sequencing.* Current protocols in molecular biology, 2014. **107**(1): p. 4.22. 1-4.22. 17.

154. Li, B. and C.N. Dewey, *RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome.* BMC bioinformatics, 2011. **12**(1): p. 1-16.

155. Lander, E.S. and M.S. Waterman, *Genomic mapping by fingerprinting random clones: a mathematical analysis.* Genomics, 1988. **2**(3): p. 231-239.

156. Leek, J.T., et al., *The sva package for removing batch effects and other unwanted variation in high-throughput experiments.* Bioinformatics, 2012. **28**(6): p. 882-883.

157. Reimand, J., et al., *Pathway enrichment analysis and visualization of omics data using g: Profiler, GSEA, Cytoscape and EnrichmentMap.* Nature protocols, 2019. **14**(2): p. 482-517.

158. Schneider, C.A., W.S. Rasband, and K.W. Eliceiri, *NIH Image to ImageJ: 25 years of image analysis.* Nature Methods, 2012. **9**(7): p. 671-675.

159. Lowe, D.G. *Object recognition from local scale-invariant features*. in *Proceedings of the seventh IEEE international conference on computer vision*. 1999. Ieee.

160. Matas, J., et al., *Robust wide-baseline stereo from maximally stable extremal regions.* Image and vision computing, 2004. **22**(10): p. 761-767.

161. Schmidhuber, J., *Deep learning in neural networks: An overview.* Neural networks, 2015. **61**: p. 85-117.

162. Rodel, H.E., et al., *Aggregated Mycobacterium tuberculosis enhances the inflammatory response.* Frontiers in microbiology, 2021. **12**.

163. Dinarello, C.A., *Overview of the IL‐1 family in innate inflammation and acquired immunity.* Immunological reviews, 2018. **281**(1): p. 8-27.

164. Baggiolini, M. and I. Clark-Lewis, *Interleukin‐8, a chemotactic and inflammatory cytokine.* FEBS letters, 1992. **307**(1): p. 97-101.

165. Menten, P., A. Wuyts, and J. Van Damme, *Macrophage inflammatory protein-1.* Cytokine & growth factor reviews, 2002. **13**(6): p. 455-481.

166. Dickinson, J.L., et al., *Plasminogen activator inhibitor type 2 inhibits tumor necrosis factor α-induced apoptosis: evidence for an alternate biological function.* Journal of Biological Chemistry, 1995. **270**(46): p. 27894-27904.

167. Danchuk, S., et al., *Human multipotent stromal cells attenuate lipopolysaccharide-induced acute lung injury in mice via secretion of tumor necrosis factor-α-induced protein 6.* Stem cell research & therapy, 2011. **2**(3): p. 1-15.

168. Church, A., et al., *Anti‐CD20 monoclonal antibody‐dependent phagocytosis of chronic lymphocytic leukaemia cells by autologous macrophages.* Clinical & Experimental Immunology, 2016. **183**(1): p. 90-101.

169. Cannon, G.J. and J.A. Swanson, *The macrophage capacity for phagocytosis.* Journal of cell science, 1992. **101**(4): p. 907-913.

170. Welin, A., et al., *Human macrophages infected with a high burden of ESAT-6-expressing M. tuberculosis undergo caspase-1-and cathepsin B-independent necrosis.* PloS one, 2011. **6**(5): p. e20302.

171. Podinovskaia, M., et al., *Infection of macrophages with Mycobacterium tuberculosis induces global modifications to phagosomal function.* Cellular Microbiology, 2013. **15**(6): p. 843-859.

172. Gleeson, L.E., et al., *Cutting edge: Mycobacterium tuberculosis induces aerobic glycolysis in human alveolar macrophages that is required for control of intracellular bacillary replication.* The Journal of Immunology, 2016. **196**(6): p. 2444-2449.

173. Man, S.M., R. Karki, and T.D. Kanneganti, *Molecular mechanisms and functions of pyroptosis, inflammatory caspases and inflammasomes in infectious diseases.* Immunological reviews, 2017. **277**(1): p. 61-75.

174. Kelley, N., et al., *The NLRP3 inflammasome: an overview of mechanisms of activation and regulation.* International journal of molecular sciences, 2019. **20**(13): p. 3328.

175. Hammond, M.E., et al., *IL-8 induces neutrophil chemotaxis predominantly via type I IL-8 receptors.* J Immunol, 1995. **155**(3): p. 1428-33.

176. Koch, A.E., et al., *Interleukin-8 as a macrophage-derived mediator of angiogenesis.* Science, 1992. **258**(5089): p. 1798-801.

177. Kobayashi, Y., *The role of chemokines in neutrophil biology.* Front Biosci, 2008. **13**(1): p. 2400-7.

178. Zhang, Y., et al., *Enhanced interleukin-8 release and gene expression in macrophages after exposure to Mycobacterium tuberculosis and its components.* J Clin Invest, 1995. **95**(2): p. 586-92.

179. Ameixa, C. and J.S. Friedland, *Interleukin-8 secretion from Mycobacterium tuberculosis-infected monocytes is regulated by protein tyrosine kinases but not by ERK1/2 or p38 mitogen-activated protein kinases.* Infect Immun, 2002. **70**(8): p. 4743-6.

180. Lowe, D.M., et al., *Neutrophils in tuberculosis: friend or foe?* Trends in immunology, 2012. **33**(1): p. 14-25.

181. Nandi, B. and S.M. Behar, *Regulation of neutrophils by interferon-γ limits lung inflammation during tuberculosis infection.* Journal of Experimental Medicine, 2011. **208**(11): p. 2251-2262.

182. Hilda, J.N., et al., *Role of neutrophils in tuberculosis: A bird's eye view.* Innate immunity, 2020. **26**(4): p. 240-247.

183. Polena, H., et al., *Mycobacterium tuberculosis exploits the formation of new blood vessels for its dissemination.* Scientific reports, 2016. **6**(1): p. 1-11.

184. Schroder, W.A., et al., *A physiological function of inflammation-associated SerpinB2 is regulation of adaptive immunity.* The Journal of Immunology, 2010. **184**(5): p. 2663-2670.

185. Parameswaran, N. and S. Patial, *Tumor necrosis factor-α signaling in macrophages.* Critical Reviews™ in Eukaryotic Gene Expression, 2010. **20**(2).

186. Kumar, S. and C. Baglioni, *Protection from tumor necrosis factor-mediated cytolysis by overexpression of plasminogen activator inhibitor type-2.* Journal of Biological Chemistry, 1991. **266**(31): p. 20960-20964.

187. Howard, O.Z., et al., *Functional redundancy of the human CCL4 and CCL4L1 chemokine genes.* Biochemical and biophysical research communications, 2004. **320**(3): p. 927-931.

188. Vesosky, B., et al., *CCL5 participates in early protection against Mycobacterium tuberculosis.* Journal of Leukocyte Biology, 2010. **87**(6): p. 1153-1165.

189. Saukkonen, J.J., et al., *Beta-chemokines are induced by Mycobacterium tuberculosis and inhibit its growth.* Infect Immun, 2002. **70**(4): p. 1684-93.

190. Kasahara, E., et al., *SOD2 protects against oxidation-induced apoptosis in mouse retinal pigment epithelium: implications for age-related macular degeneration.* Invest Ophthalmol Vis Sci, 2005. **46**(9): p. 3426-34.

191. Zhang, Y. and J. Xu, *MiR-140-5p regulates hypoxia-mediated human pulmonary artery smooth muscle cell proliferation, apoptosis and differentiation by targeting Dnmt1 and promoting SOD2 expression.* Biochem Biophys Res Commun, 2016. **473**(1): p. 342-348.

192. Milner, C.M. and A.J. Day, *TSG-6: a multifunctional protein associated with inflammation.* Journal of cell science, 2003. **116**(10): p. 1863-1873.

193. Dyer, D.P., et al., *The Anti-inflammatory Protein TSG-6 Regulates Chemokine Function by Inhibiting Chemokine/Glycosaminoglycan Interactions.* J Biol Chem, 2016. **291**(24): p. 12627-12640.

194. Wang, S., et al., *Tumor necrosis factor-inducible gene 6 protein ameliorates chronic liver damage by promoting autophagy formation in mice.* Experimental & molecular medicine, 2017. **49**(9): p. e380-e380.

195. Wu, M.X., et al., *IEX-1L, an apoptosis inhibitor involved in NF-κB-mediated cell survival.* Science, 1998. **281**(5379): p. 998-1001.

196. Bhattacharya, B., et al., *The integrated stress response mediates necrosis in murine Mycobacterium tuberculosis granulomas.* J Clin Invest, 2021. **131**(3).

197. Adankwah, E., et al., *Lower IL-7 receptor expression of monocytes impairs antimycobacterial effector functions in patients with tuberculosis.* The Journal of Immunology, 2021. **206**(10): p. 2430-2440.

198. Leung, G., C. Valencia, and A. Beaudin, *IL7R regulates fetal tissue resident macrophage development by facilitating cell survival*. 2020, Am Assoc Immnol.

199. Tien, A.L., et al., *UHRF1 depletion causes a G2/M arrest, activation of DNA damage response and apoptosis.* Biochemical Journal, 2011. **435**(1): p. 175-185.

200. Law, K., et al., *Increased release of interleukin-1 beta, interleukin-6, and tumor necrosis factor-alpha by bronchoalveolar cells lavaged from involved sites in pulmonary tuberculosis.* American journal of respiratory and critical care medicine, 1996. **153**(2): p. 799-804.

201. Ashenafi, S., et al., *Progression of clinical tuberculosis is associated with a Th2 immune response signature in combination with elevated levels of SOCS3.* Clinical immunology, 2014. **151**(2): p. 84-99.

202. Flynn, J.L., et al., *Tumor necrosis factor-alpha is required in the protective immune response against Mycobacterium tuberculosis in mice.* Immunity, 1995. **2**(6): p. 561-72.

203. O'Garra, A., et al., *The immune response in tuberculosis.* Annual review of immunology, 2013. **31**: p. 475-527.

204. Kaneko, H., et al., *Role of tumor necrosis factor-alpha in Mycobacterium-induced granuloma formation in tumor necrosis factor-alpha-deficient mice.* Laboratory investigation; a journal of technical methods and pathology, 1999. **79**(4): p. 379-386.

205. Bean, A.G., et al., *Structural deficiencies in granuloma formation in TNF gene-targeted mice underlie the heightened susceptibility to aerosol Mycobacterium tuberculosis infection, which is not compensated for by lymphotoxin.* The Journal of Immunology, 1999. **162**(6): p. 3504-3511.

206. Clay, H., H.E. Volkman, and L. Ramakrishnan, *Tumor necrosis factor signaling mediates resistance to mycobacteria by inhibiting bacterial growth and macrophage death.* Immunity, 2008. **29**(2): p. 283-294.

207. Cooper, A.M., K.D. Mayer-Barber, and A. Sher, *Role of innate cytokines in mycobacterial infection.* Mucosal immunology, 2011. **4**(3): p. 252-260.

208. Kostura, M.J., et al., *Identification of a monocyte specific pre-interleukin 1 beta convertase activity.* Proceedings of the National Academy of Sciences, 1989. **86**(14): p. 5227-5231.

209. Martinon, F., K. Burns, and J. Tschopp, *The inflammasome: a molecular platform triggering activation of inflammatory caspases and processing of proIL-β.* Molecular cell, 2002. **10**(2): p. 417-426.

210. Warny, M. and C.P. Kelly, *Monocytic cell necrosis is mediated by potassium depletion and caspase-like proteases.* Am J Physiol, 1999. **276**(3 Pt 1): p. C717-24.

211. den Hartigh, A.B. and S.L. Fink, *Pyroptosis induction and detection.* Current protocols in immunology, 2018. **122**(1): p. e52.

212. Pethe, K., et al., *Isolation of Mycobacterium tuberculosis mutants defective in the arrest of phagosome maturation.* Proceedings of the National Academy of Sciences, 2004. **101**(37): p. 13642-13647.

213. Tan, S., R.M. Yates, and D.G. Russell, *Mycobacterium tuberculosis: Readouts of Bacterial Fitness and the Environment Within the Phagosome.* Methods in molecular biology (Clifton, N.J.), 2017. **1519**: p. 333-347.

214. Deretic, V., et al., *Mycobacterium tuberculosis inhibition of phagolysosome biogenesis and autophagy as a host defence mechanism.* Cellular microbiology, 2006. **8**(5): p. 719-727.

215. McDonough, K.A., Y. Kress, and B.R. Bloom, *Pathogenesis of tuberculosis: interaction of Mycobacterium tuberculosis with macrophages.* Infect Immun, 1993. **61**(7): p. 2763-73.

216. VanderVen, B.C., R.M. Yates, and D.G. Russell, *Intraphagosomal measurement of the magnitude and duration of the oxidative burst.* Traffic, 2009. **10**(4): p. 372-378.

217. Wong, K.-W., *The role of ESX-1 in Mycobacterium tuberculosis pathogenesis.* Microbiology spectrum, 2017. **5**(3): p. 5.3. 02.

218. Sreejit, G., et al., *The ESAT-6 protein of Mycobacterium tuberculosis interacts with beta-2-microglobulin (β2M) affecting antigen presentation function of macrophage.* PLoS pathogens, 2014. **10**(10): p. e1004446.

219. Beatty, W.L., et al., *Trafficking and release of mycobacterial lipids from infected macrophages.* Traffic, 2000. **1**(3): p. 235-247.

220. Emmert-Streib, F., et al., *An introductory review of deep learning for prediction models with big data.* Frontiers in Artificial Intelligence, 2020. **3**: p. 4.

221. Mahesh, B., *Machine learning algorithms-a review.* International Journal of Science and Research (IJSR).[Internet], 2020. **9**: p. 381-386.

222. Mazurowski, M.A., et al., *Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance.* Neural networks, 2008. **21**(2-3): p. 427-436.

223. Weiss, G.M. and F. Provost, *The effect of class distribution on classifier learning: an empirical study*. 2001, Rutgers University.

224. Chen, X.-W. and X. Lin, *Big data deep learning: challenges and perspectives.* IEEE access, 2014. **2**: p. 514-525.

225. Saito, T. and M. Rehmsmeier, *The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets.* PLOS ONE, 2015. **10**(3): p. e0118432.

226. Krombach, F., et al., *Cell size of alveolar macrophages: an interspecies comparison.* Environmental health perspectives, 1997. **105**(suppl 5): p. 1261-1263.

227. Hayden, M.S. and S. Ghosh. *Regulation of NF-κB by TNF family cytokines*. in *Seminars in immunology*. 2014. Elsevier.

228. Queval, C.J., et al., *STAT3 represses nitric oxide synthesis in human macrophages upon Mycobacterium tuberculosis infection.* Scientific reports, 2016. **6**(1): p. 1-14.

229. Rex, J., et al., *IL-1β and TNFα differentially influence NF-κB activity and FasL-induced apoptosis in primary murine hepatocytes during LPS-induced inflammation.* Frontiers in physiology, 2019. **10**: p. 117.

230. Althnian, A., et al., *Impact of dataset size on classification performance: an empirical evaluation in the medical domain.* Applied Sciences, 2021. **11**(2): p. 796.

8 Chapter 8. Appendices

8.1 Supplementary Material

Supplementary table 8.1.1: DESeq2 differentially regulated genes between infection conditions at adjusted p-value < 0.01. Adapted w/o permission from Rodel *et al.* 2021 [162]

Gene	Mean reads	$\log_2$ Fold Change	Adjusted p
	Aggregate relative to Uninfected		
CCL4L1	117	6,29	<0,0001
CCL4L2	117	6,29	<0,0001
IL1B	923	5,06	<0,0001
IL8	1886	6,16	<0,0001
CCL4	374	5,96	<0,0001
SOD2	847	2,58	<0,0001
SERPINB2	53	6,39	<0,0001
TNFAIP6	232	5,40	<0,0001
CCL3	1098	2,82	<0,0001
ICAM1	702	2,04	<0,0001
HSPA1A	64	2,32	<0,0001
EHD1	206	2,03	<0,0001
CCL20	43	6,62	<0,0001
ZC3H12A	73	2,58	<0,0001
CXCL3	175	5,30	<0,0001
MFSD2A	140	1,67	<0,0001
PIM1	346	1,28	<0,0001
SLC2A6	170	1,93	<0,0001
INHBA	42	3,14	<0,0001
BTG2	316	1,10	<0,0001
OSGIN1	85	1,41	<0,0001
AMPD3	126	1,60	<0,0001

TNF	172	4,60	<0,0001
C15orf48	557	1,28	<0,0001
CXCL2	219	4,97	<0,0001
NFE2L2	300	0,81	<0,0001
CCL3L1	276	8,59	<0,0001
CXCL1	85	5,35	<0,0001
CD274	125	1,51	<0,0001
IER3	65	2,77	<0,0001
PTGS2	21	3,78	<0,0001
NCK2	16	2,40	<0,0001
NAMPT	154	1,64	<0,0001
TRAF1	80	1,69	<0,0001
LIF	31	2,06	<0,0001
HLA-DQA1	345	1,48	<0,0001
SQSTM1	2922	0,88	<0,0001
IRAK2	41	2,19	<0,0001
BID	205	0,98	<0,0001
IL7R	70	2,79	<0,0001
ADORA2A	14	4,73	<0,0001
IL6	29	5,63	<0,0001
KYNU	455	1,17	<0,0001
PDE4DIP	315	0,72	<0,0001
KDM6B	9	2,89	<0,0001
MAP3K8	26	1,50	<0,0001
RNF144B	37	1,70	<0,0001
DRAM1	166	1,04	<0,0001
CSF1	437	1,13	<0,001
MSANTD3	30	1,49	<0,001
MSC	305	0,93	<0,001
AMZ1	68	1,61	<0,001
G0S2	30	2,75	<0,001
GPR35	18	2,60	<0,001

TNFAIP3	151	1,84	<0,001
TRAF3	69	1,17	<0,001
MTHFD2	446	0,65	<0,001
HCK	309	0,64	<0,001
RILPL2	94	0,91	<0,001
HSPA1B	19	2,26	<0,001
SLC2A3	231	1,14	<0,001
ATP2B1	55	1,24	<0,001
GEM	70	1,08	<0,001
HMOX1	573	0,63	<0,01
B4GALT1	391	1,05	<0,01
PPP1R15A	444	0,96	<0,01
IFIT2	27	1,72	<0,01
STX11	114	0,79	<0,01
MAP2K3	525	0,71	<0,01
CD80	4	3,03	<0,01
CES1	237	1,54	<0,01
RIN2	30	-1,39	<0,01
TREM1	37	1,94	<0,01
IL1A	26	2,66	<0,01
CCRL2	122	0,87	<0,01
CD180	41	-1,27	<0,01
PDK4	126	-1,15	<0,01
CYP27B1	63	1,01	<0,01
TNFSF15	45	1,96	<0,01
EDN1	6	3,27	<0,01
SDC4	239	0,71	<0,01
HIVEP2	21	1,85	<0,01
NFKBIA	397	1,26	<0,01
CD83	972	0,72	<0,01
LINC00674	35	-0,99	<0,01
DENND5A	43	1,23	<0,01

USF1	19	1,15	<0,01
MORC3	62	0,96	<0,01
	Multiple relative to Uninfected		
IL1B	923	4,33	<0,0001
SERPINB2	53	6,27	<0,0001
SOD2	847	2,40	<0,0001
CCL4L1	117	4,81	<0,0001
CCL4L2	117	4,81	<0,0001
IL8	1886	5,06	<0,0001
TNFAIP6	232	5,01	<0,0001
SLC2A6	170	2,16	<0,0001
CCL4	374	4,26	<0,0001
C15orf48	557	1,49	<0,0001
PIM1	346	1,28	<0,0001
BTG2	316	1,11	<0,0001
EHD1	206	1,73	<0,0001
AMPD3	126	1,54	<0,0001
HCK	309	0,87	<0,0001
BID	205	1,08	<0,0001
IL7R	70	3,08	<0,0001
CCL3	1098	1,87	<0,0001
ICAM1	702	1,44	<0,0001
OSGIN1	85	1,22	<0,0001
SQSTM1	2922	0,85	<0,0001
CXCL1	85	4,42	<0,0001
TNF	172	3,73	<0,0001
CD274	125	1,30	<0,0001
CXCL3	175	3,54	<0,0001
KYNU	455	1,13	<0,0001
GPR35	18	2,68	<0,001
CCL20	43	4,16	<0,001
NCK2	16	1,90	<0,001

PDE4DIP	315	0,63	<0,001
NAMPT	154	1,25	<0,001
ACSL1	2158	1,30	<0,01
TRAF1	80	1,29	<0,01
NCF1	30	1,74	<0,01
MSC	305	0,83	<0,01
CCL3L1	276	5,57	<0,01
INHBA	42	1,90	<0,01
HLA-DQA1	345	1,09	<0,01
DENND5A	43	1,31	<0,01
NCAPH	192	-0,81	<0,01
SLC7A11	398	1,11	<0,01
NFE2L2	300	0,50	<0,01
DRAM1	166	0,85	<0,01
IFNGR2	275	0,93	<0,01
RILPL2	94	0,78	<0,01
CES1	237	1,43	<0,01
MTHFD2	446	0,55	<0,01
GPR68	53	1,09	<0,01
AMZ1	68	1,35	<0,01
MFSD2A	140	0,91	<0,01
IVNS1ABP	435	-0,52	<0,01
	Single relative to Uninfected		
SERPINB2	53	5,63	<0,0001
TNFAIP6	232	4,83	<0,0001
SOD2	847	1,94	<0,0001
IL1B	923	3,27	<0,0001
CCL4L1	117	3,75	<0,0001
CCL4L2	117	3,75	<0,0001
IL8	1886	3,85	<0,0001
SLC2A6	170	1,71	<0,0001
PIM1	346	1,09	<0,0001

AMPD3	126	1,39	<0,0001
EHD1	206	1,47	<0,0001
C15orf48	557	1,05	<0,0001
CCL4	374	2,94	<0,0001
KYNU	455	1,23	<0,0001
BTG2	316	0,82	<0,0001
ICAM1	702	1,24	<0,0001
BID	205	0,87	<0,001
HCK	309	0,67	<0,001
SQSTM1	2922	0,74	<0,001
OSGIN1	85	1,01	<0,001
TNIP1	38	1,40	<0,001
MSC	305	0,86	<0,01
CCL3	1098	1,28	<0,01
PDE4DIP	315	0,57	<0,01
TRAF1	80	1,20	<0,01
HMOX1	573	0,58	<0,01
	Aggregate relative to Single		
CXCL2	219	4,77	<0,0001
CCL4	374	3,03	<0,0001
CCL4L1	117	2,54	<0,0001
CCL4L2	117	2,54	<0,0001
TNF	172	3,97	<0,0001
HSPA1A	64	1,67	<0,0001
CXCL3	175	3,63	<0,0001
ZC3H12A	73	1,68	<0,001
CCL3	1098	1,54	<0,001
IER3	65	2,20	<0,001
MAP3K8	26	1,43	<0,001
CCL2	103	1,40	<0,01
CXCL1	85	3,55	<0,01
IRAK2	41	1,76	<0,01

INHBA	42	1,86	<0,01
IL8	1886	2,31	<0,01
	Aggregate relative to Multiple		
HSPA1A	64	1,51	<0,001

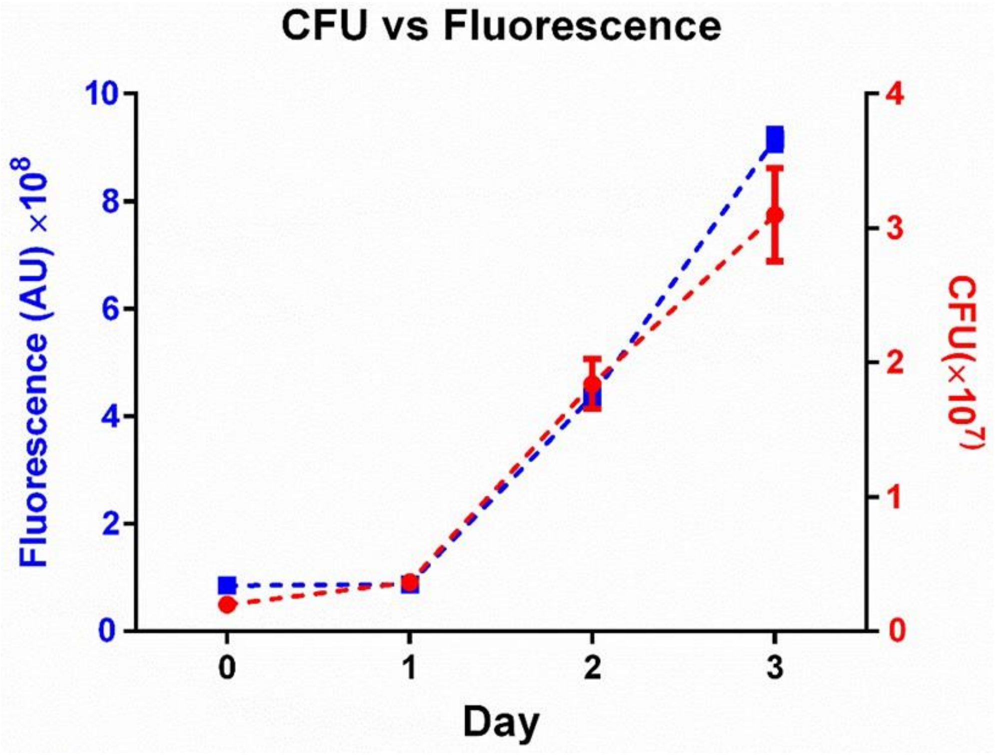


Figure 8.1: Mtb fluorescence measures bacterial numbers. Adapted w/o permission from Mahamed *et al.* 2017 [30]

Mtb colony forming units (CFU – red circles) versus Mtb fluorescence (blue squares) tracked over 3 days of growth in culture (mean ±sd).

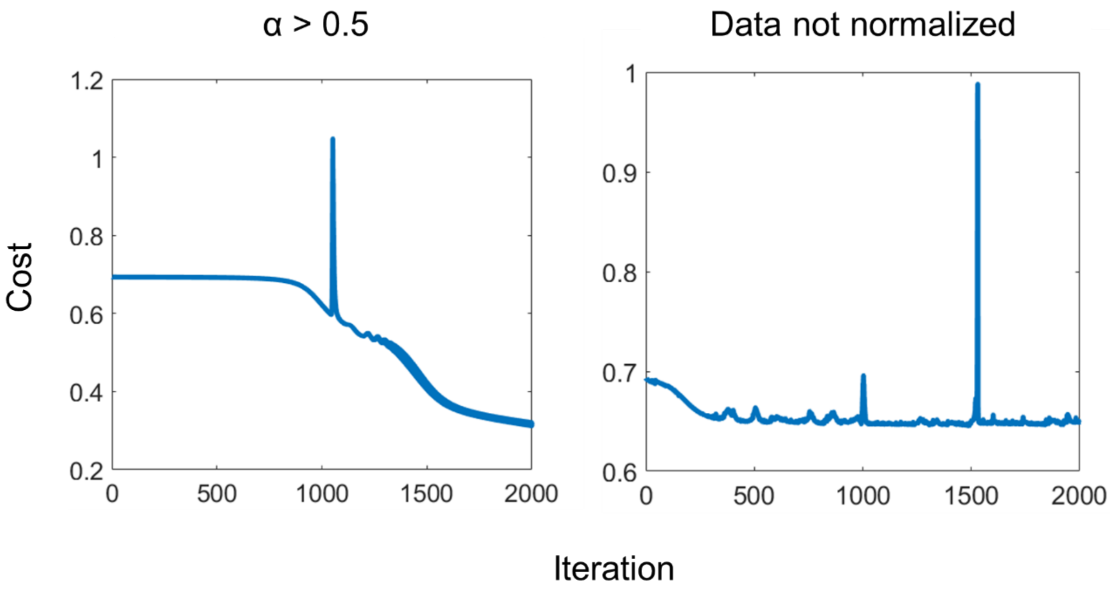


Figure 8.2: Hyperparameter calibration and data normalization is important for predictable CNN training

Large values of alpha (left panel) and training data that has not been normalized (right panel) can result in unpredictable CNN training and result in poor model performance. Y axis is cost (model error), X axis is training iteration number

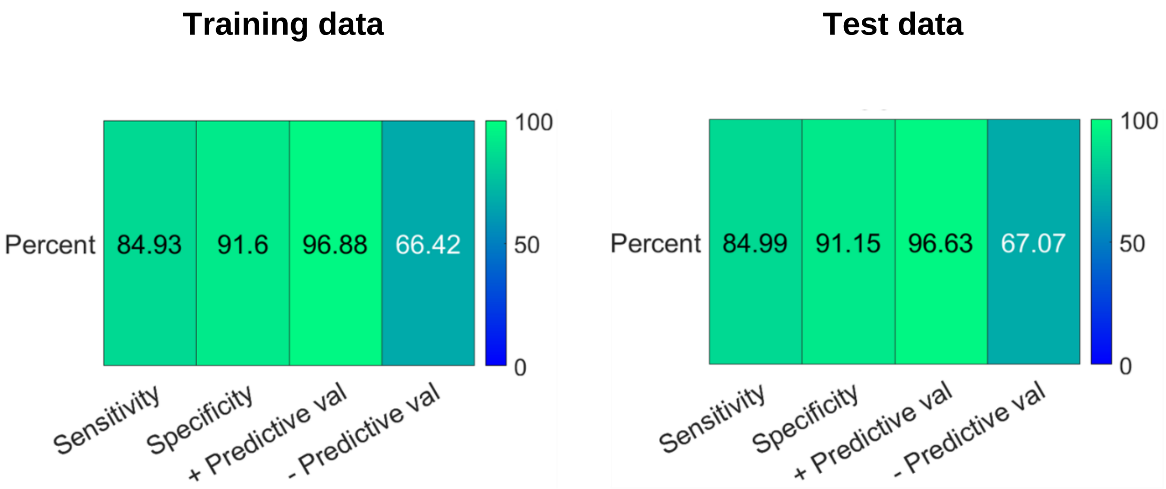


Figure 8.3: Neuron 3 of HyRoNet is optimized for positive predictive value

Performance of CNN neuron 3 when applied directly to data resulting from classification by neuron 2 of HyRoNet for training (left panel) and cross validation (right panel) datasets. Colour in scale bar indicates percentage.

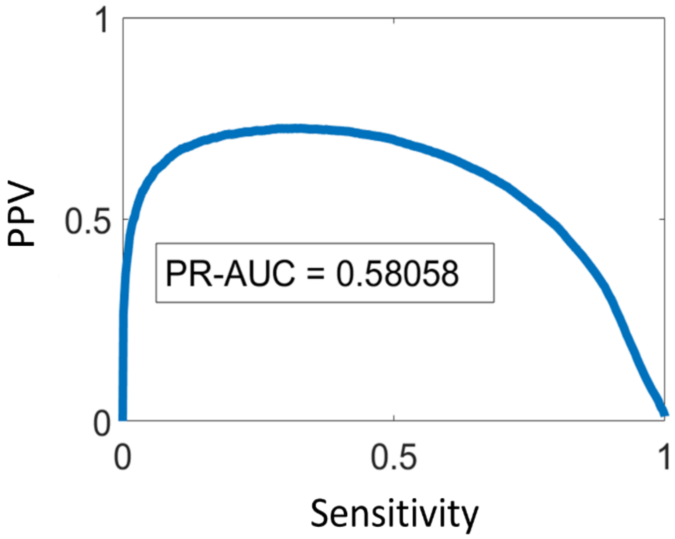


Figure 8.4: Single neuron CNN architecture has low precision recall

A CNN architecture using only a single neuron for classification has low sensitivity and positive predictive value (measured using precision recall area under curve: PR-AUC).

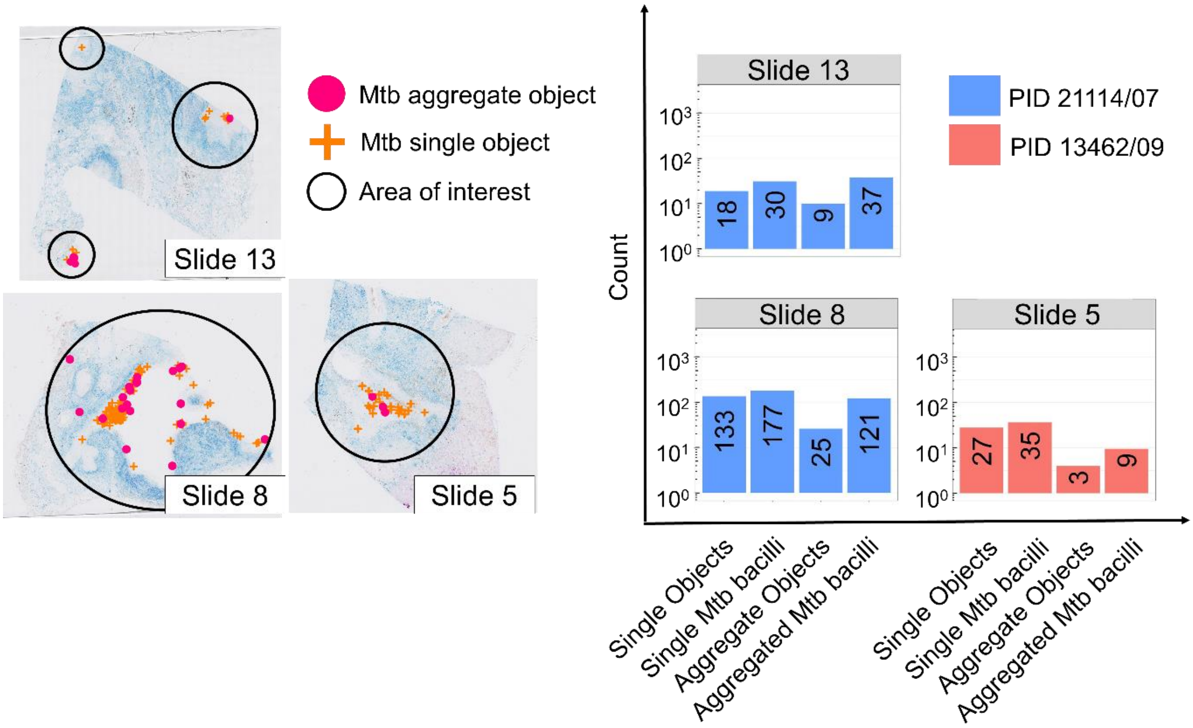


Figure 8.5: Quantification of Mtb in duplicated slides

Tissue slides indicate locations of Mtb singles (orange crosses) and Mtb aggregates (magenta circles) within areas circled in black. Bar graphs show quantification data for Mtb singlet and aggregate objects and the estimated number of bacilli in the corresponding tissue slide. Bar colour indicates slide PID.

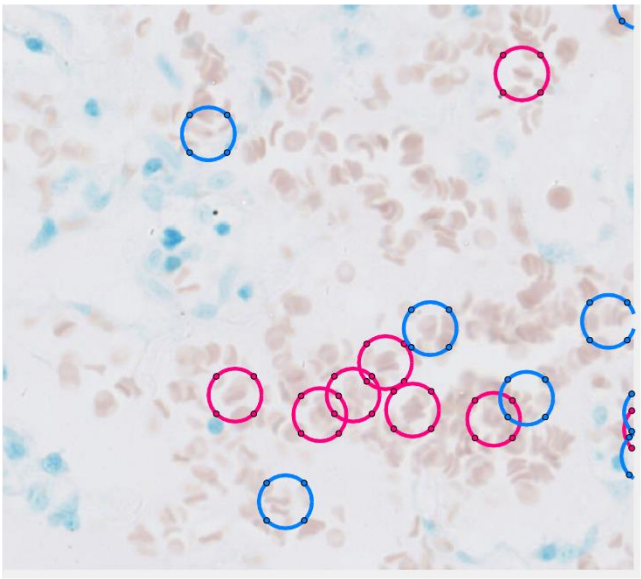


Figure 8.6: RBC false positives found during validation of HyRoNet classifications

Many false positives detected by the HyRoNet CNN were identified as red blood cells during manual validation.

8.2 Code appendix

All executable code segments are contained in beige text blocks. Code comments are written in green text and preceded by one or more percentage symbols (%Example comment). String variables are written in red text and surrounded by single or double apostrophe symbols (‘Example string’). Loop or function operators are defined by blue text (Example loop). All other code is in plain black text.

8.2.1 Feature Extractor

Mandatory user input parameters are defined in this block of code.

```
%Enter file name of high magnification image (40x)
File_name = ‘Slide name string.tif';

%Enter file name of low magnifications image (0.625x).
File_name_small = ‘low mag file.tif';

%Define minimum pixel size for small object removal
bckG_pxl_sz = 20;

%Define minimum pixel size for small object removal in non-target database
bckG_pxl_sz_2 = 5;

%Define minimum area (in microns)
Micron_limit = 1.5;

%Define background threshold in standard deviations from the mean, for
%Magenta-biased BW mask.
Std_DevThresh_Multiplier = 2;

%Define background threshold, in standard deviations from the mean, for
%Full RGB spectrum
```

```
Std_DevThresh_MultiplierV2 = 0.75;

%Define background threshold, in standard deviations from the mean, for
%Red-biased BW mask.
Std_DevThresh_MultiplierV3 = 0.75;

% Define background threshold, in standard deviations from the mean, for
%Blue-biased BW mask.
Std_DevThresh_MultiplierV4 = 0.75;

%Set background thresholds, in standard deviation from the mean, for non-
%target database
BrightThresh = 1.5;
DarkThresh = 4.5;
BYelThresh = 3;
RGThresh = 1.5;

%Define Mtb aggregate area in microns
Mtb_sz_val = 6.8;

%Size of circles indicating Mtb positions in tissue slide
circSize = 150;

%Define name of output training dataset file after feature extraction and
validation
DataSetName = ‘Dataset name’;
```

Large image file in .tiff file format are split into smaller tiles and loaded into the MATLAB® workspace for processing

```
%%This section of code identifies the dimensions of the large scanned
%%image and subdivides it into manageable tiles for analysis. It also creates
%%a tile number system so that specific tiles can be located later in the
```

```
%%%script

%Initialize variables
Total_Objects = [];
Total_Not_Objects = [];

%Get image metadata
FullInfo = imfinfo(File_name);
FullRows = FullInfo.Height;
FullColumns = FullInfo.Width;

%Define range of divisors for image split
DivisorS = 15:35;

%Find image divisor (in pixels) without remainder for image length and width
RowMat = ones(size(DivisorS)) * FullRows;
RowModuluS = RowMat./DivisorS;
RowDivisorsPxlSz = cat(1,DivisorS(~logical(mod(RowModuluS,1))),RowModuluS(~logical(mod(RowModuluS,1))));
[~, mIndex] = min(abs((5000 - RowDivisorsPxlSz(2,:))));
RowDivisorsPxlSz = RowDivisorsPxlSz(:,mIndex);
ColMat = ones(size(DivisorS)) * FullColumns;
ColModuluS = ColMat./DivisorS;
ColDivisorsPxlSz = cat(1,DivisorS(~logical(mod(ColModuluS,1))),ColModuluS(~logical(mod(ColModuluS,1))));
[~, mIndex] = min(abs((5000 - ColDivisorsPxlSz(2,:))));
ColDivisorsPxlSz = ColDivisorsPxlSz(:,mIndex);
Mosaic_Dim = [RowDivisorsPxlSz(1) ColDivisorsPxlSz(1)];
Tile_Dim = [RowDivisorsPxlSz(2) ColDivisorsPxlSz(2)];

%Create image matrix indexing vectors
```

```
RowInd1 = 0:Tile_Dim(1):FullRows;
RowInd1 = RowInd1(2:end);
RowInd2 = 1:Tile_Dim(1):FullRows;
RowIndX = [RowInd2; RowInd1]';
ColInd1 = 0:Tile_Dim(2):FullColumns;
ColInd1 = ColInd1(2:end);
ColInd2 = 1:Tile_Dim(2):FullColumns;
ColIndX = [ColInd2; ColInd1]';

%Initialize empty tiling index variable
TileIndex = [];
```

Image tiles are individually loaded for processing and feature extraction

```
%Begin feature extraction loop
for vv = 1:size(RowIndX,1)
   for ww = 1:size(ColIndX,1)
%Select image region (tile) for processing
        Tileflag = ((vv-1) * size(ColIndX,1) + (ww-1))+1;
        TileIndex =
[TileIndex;RowIndX(vv,1),RowIndX(vv,2),ColIndX(ww,1),ColIndX(ww,2),Tileflag];
        ['Analysis '
num2str(round((Tileflag/(Mosaic_Dim(1)*Mosaic_Dim(2))*100))) '%
complete']
        one_res_level =
double(imread(File_name,'PixelRegion',{[RowIndX(vv,1),RowIndX(vv,2)],[ColIndX(ww,1),ColIndX(ww,2)]}));
%Extract Hue, Saturation and Value matrices
        HSV = rgb2hsv(one_res_level);
                H = HSV(:,:,1);
                S = HSV(:,:,2);
                V = HSV(:,:,3);
%Split Whole RGB image into individual grayscale intensity matrices
        redLayer = (one_res_level(:,:,1));
        greenLayer = (one_res_level(:,:,2));
        blueLayer = (one_res_level(:,:,3));
        blank = double(zeros(size(redLayer)));
%Construct additional grayscale intensity matrices
        MagentaMap = ((redLayer+blueLayer) - greenLayer);
        CyanMap = ((blueLayer+greenLayer) - redLayer);
        YellowMap = ((redLayer + greenLayer) - blueLayer);
        lightmap = ((redLayer+blueLayer+greenLayer)./3)+1;
        Darkmap = imcomplement(lightmap);
        Darkmap = Darkmap + abs(min(min(Darkmap)));
%Construct bias matrices for target (Ziehl Neelson) and non-target RGB
%profiles
        Mask1 = MagentaMap - greenLayer;
        Mask1 = Mask1+(abs(min(min(Mask1))));
        Mask1V2 = ((MagentaMap - greenLayer)./lightmap).*255;
        Mask1V2 = Mask1V2+(abs(min(min(Mask1V2))));
        Mask1V3 = redLayer-CyanMap-CyanMap;
        Mask1V3 = Mask1V3+(abs(min(min(Mask1V3))));
        MeanMaskV3 = mean(mean(Mask1V3));
        stdMaskV3 = std(Mask1V3(Mask1V3>0),0,[1 2]);
        Mask1V3B = Mask1V3 -
(MeanMaskV3+(Std_DevThresh_MultiplierV3*stdMaskV3));
        Mask3V3 = Mask1V3B>0;
        Mask1V4 = blueLayer-YellowMap-YellowMap;
        Mask1V4 = Mask1V4+(abs(min(min(Mask1V4))));
        MeanMaskV4 = mean(mean(Mask1V4));
        stdMaskV4 = std(Mask1V4(Mask1V4>0),0,[1 2]);
        Mask1V4B = Mask1V4 -
(MeanMaskV4+(Std_DevThresh_MultiplierV4*stdMaskV4));
```

```
        Mask3V4 = Mask1V3B>0;
        BY = blueLayer - (redLayer+greenLayer);
        BY = BY + abs(min(min(BY)));
        RG = imcomplement(redLayer-greenLayer);
        RG = RG + abs(min(min(RG)));
%Apply low, user defined,, thresholds to bias matrices
        MeanMask = mean(mean(Mask1));
        stdMask = std(Mask1(Mask1>0),0,[1 2]);
        Mask1B = Mask1 - (MeanMask+(Std_DevThresh_Multiplier*stdMask));
        Mask3 = Mask1B>0;
        MeanMaskV2 = mean(mean(Mask1V2));
        stdMaskV2 = std(Mask1V2(Mask1V2>0),0,[1 2]);
        Mask1V2B = Mask1V2 -
(MeanMaskV2+(Std_DevThresh_MultiplierV2*stdMaskV2));
        Mask3V2 = Mask1V2B>0;
%Apply Boolean intersection on target bias matrices
       Mask3 = bwareaopen(Mask3 & Mask3V2 & Mask3V3 &
Mask3V4,bckG_pxl_sz);
%Apply Boolean intersection on non-target matrices
        lightmap(~Mask3) = 0;
        Darkmap(~Mask3) = 0;
        BY(~Mask3) = 0;
        RG(~Mask3) = 0;
        MeanB = range(range(lightmap))/2;
        stdB = abs(std(lightmap(lightmap>0),0,[1 2]));
        lightmap(lightmap<(MeanB+(BrightThresh*stdB))) = 0;
        lightmap = lightmap>0;
        MeanD = range(range(Darkmap))/2;
        stdD = abs(std(Darkmap(Darkmap>0),0,[1 2]));
        Darkmap(Darkmap<(MeanD+(DarkThresh*stdD))) = 0;
        Darkmap = Darkmap>0;
        MeanBY = range(range(BY))/2;
        stdBY = abs(std(BY(BY>0),0,[1 2]));
        BY(BY<(MeanBY+(BYelThresh*stdBY))) = 0;
        BY = BY>0;
        MeanRG = range(range(RG))/2;
        stdRG = abs(std(RG(RG>0),0,[1 2]));
        RG(RG<(MeanRG+(RGThresh*stdRG))) = 0;
        RG = RG>0;
        NotTargetMask = lightmap + Darkmap + BY + RG;
        NotTargetMask = NotTargetMask>0;
        RGB_NotMtb_Mask = bwareaopen(NotTargetMask ,bckG_pxl_sz_2);
        RGB_Mtb_Mask = Mask3;
        RGB_Mtb_Mask2 = imclearborder(RGB_Mtb_Mask,4);
%Convert seprate pixels into objects (target and non-target)
        CC_NotMtb = bwconncomp(NOTMtb_Mask,4);
        CC = bwconncomp(RGB_Mtb_Mask2,4);
        centroids_Mtb = regionprops(CC,
'Centroid','PixelIdxList','Area','Circularity','Eccentricity','Solidity','BoundingBox'
);
        centroids_Not_Mtb = regionprops(CC_NotMtb,
'Centroid','PixelIdxList','Area','Circularity','Eccentricity','Solidity');

        if numel(centroids_Mtb)<1 == 1
            continue
        else
```

Extract features for Mtb objects, add tile Index field and calculate size in square microns

```
            resUnit = FullInfo.ResolutionUnit;

                if strncmp(resUnit,'Cen',3) == 1
                    cnvFact = 10000;
                elseif strncmp(resUnit,'Mil',3) == 1
            cnvFact = 1000;
        else
          error('conversion units do not match')
        end
%Calculate individual object size in square microns using image metadata
      XMicron_perPixel = 1/(FullInfo.XResolution/cnvFact);
      YMicron_perPixel = 1/(FullInfo.YResolution/cnvFact);
      Square_microns_per_pixel = XMicron_perPixel*YMicron_perPixel;
        for gg = 1:numel(centroids_Mtb)
          centroids_Mtb(gg).Sqr_microns =
centroids_Mtb(gg).Area*Square_microns_per_pixel;
        end
      centroids_Mtb([centroids_Mtb.Sqr_microns] < Micron_limit ) = [];

        if numel(centroids_Mtb)<1 == 1
          continue
        else
%Extract individual object RGB means
          for jj = 1:numel(centroids_Mtb)
          centroids_Mtb(jj).Tile_number = Tileflag;
          centroids_Mtb(jj).Object_ID =
str2double([num2str(floor(log10(Tileflag))) num2str(Tileflag) num2str(jj)]);
          centroids_Mtb(jj).RGB_RedMean =
mean(redLayer(centroids_Mtb(jj).PixelIdxList));
          centroids_Mtb(jj).RGB_GreenMean =
mean(greenLayer(centroids_Mtb(jj).PixelIdxList));
          centroids_Mtb(jj).RGB_BlueMean =
mean(blueLayer(centroids_Mtb(jj).PixelIdxList));
          centroids_Mtb(jj).RGB_RedSum =
sum(sum(redLayer(centroids_Mtb(jj).PixelIdxList)));
          centroids_Mtb(jj).RGB_GreenSum =
sum(sum(greenLayer(centroids_Mtb(jj).PixelIdxList)));
          centroids_Mtb(jj).RGB_BlueSum =
sum(sum(blueLayer(centroids_Mtb(jj).PixelIdxList)));
%Extract individual object RGB max values
            centroids_Mtb(jj).RGB_RedMax =
max(max(redLayer(centroids_Mtb(jj).PixelIdxList)));
            centroids_Mtb(jj).RGB_GreenMax =
max(max(greenLayer(centroids_Mtb(jj).PixelIdxList)));
            centroids_Mtb(jj).RGB_BlueMax =
max(max(blueLayer(centroids_Mtb(jj).PixelIdxList)));
%Extract individual object RGB standard deviations
            centroids_Mtb(jj).RGB_RedStd =
std((redLayer(centroids_Mtb(jj).PixelIdxList)),0,"all");
            centroids_Mtb(jj).RGB_GreenStd =
std((greenLayer(centroids_Mtb(jj).PixelIdxList)),0,"all");
            centroids_Mtb(jj).RGB_BlueStd =
std((blueLayer(centroids_Mtb(jj).PixelIdxList)),0,"all");
%Extract individual object RGB modes
            centroids_Mtb(jj).RGB_RedMode =
mode(floor((redLayer(centroids_Mtb(jj).PixelIdxList))),"all");
            centroids_Mtb(jj).RGB_GreenMode =
mode(floor((greenLayer(centroids_Mtb(jj).PixelIdxList))),"all");
            centroids_Mtb(jj).RGB_BlueMode =
mode(floor((blueLayer(centroids_Mtb(jj).PixelIdxList))),"all");
%Extract individual object HSV means
            centroids_Mtb(jj).HSV_H =
mean(H(centroids_Mtb(jj).PixelIdxList));
            centroids_Mtb(jj).HSV_S =
mean(S(centroids_Mtb(jj).PixelIdxList));
            centroids_Mtb(jj).HSV_V =
mean(V(centroids_Mtb(jj).PixelIdxList));
%Extract individual object SIFT and MSER features from bounding box
            blank(centroids_Mtb(jj).PixelIdxList) =
redLayer(centroids_Mtb(jj).PixelIdxList);
            HOGidx = floor(centroids_Mtb(jj).BoundingBox);
```

```
          HOGimg =
blank([HOGidx(2):(HOGidx(2)+HOGidx(4))],[HOGidx(1):(HOGidx(1)+HOGidx
(3))]);
          if size(HOGimg,1)>3 && size(HOGimg,2)>3
            centroids_Mtb(jj).MSER = detectMSERFeatures(HOGimg);
            centroids_Mtb(jj).MSER = size(centroids_Mtb(jj).MSER,1);
          else
            centroids_Mtb(jj).MSER = 0;
          end
          centroids_Mtb(jj).SIFT = detectSIFTFeatures(HOGimg);
          if isempty(centroids_Mtb(jj).SIFT) == 0
            centroids_Mtb(jj).SIFTMet =
sum(centroids_Mtb(jj).SIFT.Metric);
          else
            centroids_Mtb(jj).SIFTMet = 0;
          end
          centroids_Mtb(jj).SIFT = size(centroids_Mtb(jj).SIFT,1);
          end
%Concatenate all object data with data from other image tiles
        Total_Objects = [Total_Objects;centroids_Mtb];

        end
    end
  end
end
```

Find mean whole slide RGB values from reduced magnification image

```
%Calculate whole tissue slide mean RGB values
bckGrnd_Cal = double(imread(File_name_small));
BckRGBVals = Background_Finder(bckGrnd_Cal);
%Add mean whole slide RGB values to each object in dataset
```

```
for tt = 1:numel(Total_Objects)
    Total_Objects(tt).BckRGBMeansR = BckRGBVals(1);
    Total_Objects(tt).BckRGBMeansG = BckRGBVals(2);
    Total_Objects(tt).BckRGBMeansB = BckRGBVals(3);

end
```

Export data to tab delimited file

```
%Remove extraneous fields and save extracted object data
Total_Objects = rmfield(Total_Objects,{'PixelIdxList'});
PreExport1 = struct2table(Total_Objects);
writetable(PreExport1,[DataSetName '_Full'])
PreExport = struct2table(rmfield(Total_Objects,{'Centroid','Tile_number','Object_ID','BoundingBox'}));
Export = table2array(PreExport);
writematrix(Export,DataSetName,'Delimiter','\t')
```

8.2.2 Slide RGB calculator

Finds the mean RGB profile of a whole tissue sections

```
function BckVect = Background_Finder(ImgX)
%Select tissue area from slide and calculate mean RGB values
A = ImgX;
redLayer = A(:,:,1);
blueLayer = A(:,:,2);
greenLayer = A(:,:,3);

B = imcomplement((redLayer + blueLayer + greenLayer)./3);
```

```
C = B + abs(min(min(B)));
MeanC = 255/5;
C(C<MeanC) = 0;
C(C>0) = 1;
redLayer(~C) = 0;
greenLayer(~C) = 0;
blueLayer(~C) = 0;
MeanR = mean(redLayer(redLayer>0));
MeanG = mean(greenLayer(redLayer>0));
MeanB = mean(blueLayer(redLayer>0));
BckVect = [MeanR MeanG MeanB];

end
```

8.2.3 Slide dataset refiner

Guided removal of large numbers of false positive objects in object databases extracted from whole slides

```
%Must be run directly after feature extraction with all variables still loaded in
%the MATLAB workspace after processing a full tissue slide.

%Add row number tag for objects in Total_Objects data structure
for ii = 1:numel(Total_Objects)
    Total_Objects(ii).Tag = ii;
end

%Extract RGB means for all objects in the Total_Objects data structure
RM = [Total_Objects.RGB_RedMean]';
GM = [Total_Objects.RGB_GreenMean]';
BM = [Total_Objects.RGB_BlueMean]';
ID = [Total_Objects.Tag]';
```

```
%Extract RGB means for all non-target objects extracted from the slide
RMNot = [Total_Not_Objects.RGB_RedMean]';
GMNot = [Total_Not_Objects.RGB_GreenMean]';
BMNot = [Total_Not_Objects.RGB_BlueMean]';

%Plot RGB means for target objects and non target objects on the same
%axes (in different colours). Use the MATLAB figure brush tool to select and
%delete target objects that overlap with non-target objects. Export the
%remaining datapoints to the “Mtb_index” variable.
scatter3(RM,GM,BM,0.5,'filled','b')
xlabel('Red')
ylabel('Green')
zlabel('Blue')
hold on
scatter3(RMNot,GMNot,BMNot,0.5,'filled','r')
hold off

%Extract objects identified in the “Mtb_index” variable from the whole slide
dataset to create new variable
catRGB_ID = [RM, GM, BM, ID];
KeepVect = [];
for ii = 1: size(catRGB_ID,1)
    ii/size(catRGB_ID,1)*100
        for jj = 1:(size(Mtb_index,1))
            if isequal(catRGB_ID(ii,1), Mtb_index(jj,1)) &&
isequal(catRGB_ID(ii,2), Mtb_index(jj,2)) && isequal(catRGB_ID(ii,3),
Mtb_index(jj,3))
                KeepVect = [KeepVect; catRGB_ID(ii,4)];
            else
            end
        end
end
KeepVect = unique(KeepVect);
```

```
Total_Ziel_Objects = Total_Objects(KeepVect);

%%Assign aggregate classification based on square micron area
for mm = 1:numel(Total_Ziel_Objects)
    if Total_Ziel_Objects(mm).Sqr_microns <= Mtb_sz_val
        Total_Ziel_Objects(mm).Size_class = [0 0.5 1];
    else
        Total_Ziel_Objects(mm).Size_class = [1 0 0.5];
    end
end

%Remove extraneous fields from data structures before saving
Total_Objects = rmfield(Total_Objects,{'PixelIdxList','BoundingBox'});
Total_Not_Objects = rmfield(Total_Not_Objects,{'PixelIdxList'});

%Save data structures
save([cd '\' DataSetName '\Total_objects_' DataSetName '.mat'],
'Total_Objects')
save([cd '\' DataSetName '\Ziel_objects_pre-curated' DataSetName '.mat'],
'Total_Ziel_Objects','TileIndex','Tileflag' )
save([cd '\' DataSetName '\Mtb_index_' DataSetName '.mat'], 'Mtb_index')
save([cd '\' DataSetName '\Total_Not_objects_' DataSetName '.mat'],
'Total_Not_Objects' )
```

Create MATLAB figures for each image tile containing at least one Mtb object. Used later for individual object validation in the context of surrounding tissue

```
%Create matlab figures of tiles containing objects with a high likelihood
%of being Ziel Neelson stained Mtb. Also creates a low resolution image of
%all image tiles stitched together with Ziel Nielson objects locations added

ObjTiles = unique([Total_Ziel_Objects.Tile_number]);
```

```
%Identify locations of Mtb singles and aggregates and plot on whole slide
image
for ii = 1:size(TileIndex,1)
    A =
imread(File_name,'PixelRegion',{[TileIndex(ii,1),TileIndex(ii,2)],[TileIndex(ii,3)
,TileIndex(ii,4)]});

        if (sum(ii == ObjTiles)) == 1
            SinBin = zeros(size(one_res_level,1),size(one_res_level,2));
            AggBin = zeros(size(one_res_level,1),size(one_res_level,2));
            one_res_level = A;
            redLayer = one_res_level(:,:,1);
            greenLayer = one_res_level(:,:,2);
            blueLayer = one_res_level(:,:,3);

%Find Mtb singles locations
            SubStruct = Total_Ziel_Objects([Total_Ziel_Objects.Tile_number] ==
ii);
            SubS = SubStruct([SubStruct.Sqr_microns] < Mtb_sz_val);
            a = [SubS.Centroid];
            b = a(1:2:end);
            c = a(2:2:end);
            d = [b',c'];
                for jj = 1:size(d,1)
                    SinBin(round(d(jj,2)),round(d(jj,1))) = 1;
                end

            SinBin = bwdist(SinBin);
            SinBin(SinBin <= circSize) = 1;
            SinBin(SinBin > circSize) = 0;
            SinBin = logical(SinBin);
```

```
%Find Mtb aggregates locations
        SubA = SubStruct([SubStruct.Sqr_microns] >= Mtb_sz_val);
        e = [SubA.Centroid];
        f = e(1:2:end);
        g = e(2:2:end);
        h = [f',g'];

           for jj = 1:size(h,1)
              AggBin(round(h(jj,2)),round(h(jj,1))) = 1;
           end
%Plot Mtb locations and save MATLAB figures
        AggBin = bwdist(AggBin);
        AggBin(AggBin <= circSize) = 1;
        AggBin(AggBin > circSize) = 0;
        AggBin = logical(AggBin);
        redLayer(SinBin) = 255;
        greenLayer(SinBin) = 127;
        blueLayer(SinBin) = 0;
        redLayer(AggBin) = 255;
        greenLayer(AggBin) = 0;
        blueLayer(AggBin) = 127;
        RGB_4Array = cat(3,redLayer,greenLayer,blueLayer);
        Tile_Rsz = imresize(RGB_4Array,0.2);
        Tile_RszBCK = imresize(one_res_level,0.2);
        Tile_Rsz = insertText(Tile_Rsz,[100,100],ii,'Fontsize',100,'BoxOpacity',0,'TextColor','black');
        Cat_img_array{ii} = Tile_Rsz;
        Cat_img_arrayBCK{ii} = Tile_RszBCK;

        bshow = imshow(A);
        hold on
           for kk = 1:numel(SubStruct)
```

```
                h = drawcircle('Center',SubStruct(kk).Centroid,...

'Radius',30,'Label',num2str(SubStruct(kk).Object_ID),'LabelVisible','hover',...
                    'color', SubStruct(kk).Size_class, 'FaceAlpha', 0,...
                    'InteractionsAllowed','reshape', 'LineWidth', 4,...
                    'Deletable',true);
                end
            save([cd '\' DataSetName '\Centroids_Mtb_' File_name(1:end-4)
'_Tile_' num2str(TileIndex(ii,5)) '.mat'], 'SubStruct');
            savefig([cd '\' DataSetName '\Figure_' File_name(1:end-4) '_Tile_'
num2str(TileIndex(ii,5)) '.fig' ]);
            close(gcf)

        else
            A =
imread(File_name,'PixelRegion',{[TileIndex(ii,1),TileIndex(ii,2)],[TileIndex(ii,3)
,TileIndex(ii,4)]});
            one_res_level = A;
            Tile_Rsz = imresize(one_res_level,0.2);
            Tile_RszBCK = imresize(one_res_level,0.2);
            Tile_Rsz =
insertText(Tile_Rsz,[100,100],ii,'Fontsize',100,'BoxOpacity',0,'TextColor','blac
k');
            Cat_img_array{ii} = Tile_Rsz;
            Cat_img_arrayBCK{ii} = Tile_RszBCK;
        end
end

%%%Create overview image of entire slide, extract mean RGB and add to
%%%Total_Objects data structure
bckGrnd_calibrateImg = montage(Cat_img_arrayBCK, 'Size', Mosaic_Dim);
bckGrnd_Cal = double(get(bckGrnd_calibrateImg, 'CData'));
close(gcf)
```

```
BckRGBVals = Background_Finder(bckGrnd_Cal);

for tt = 1:numel(Total_Objects)

    Total_Objects(tt).BckRGBMeansR = BckRGBVals(1);
    Total_Objects(tt).BckRGBMeansG = BckRGBVals(2);
    Total_Objects(tt).BckRGBMeansB = BckRGBVals(3);

end

for tt = 1:numel(Total_Ziel_Objects)

    Total_Ziel_Objects(tt).BckRGBMeansR = BckRGBVals(1);
    Total_Ziel_Objects(tt).BckRGBMeansG = BckRGBVals(2);
    Total_Ziel_Objects(tt).BckRGBMeansB = BckRGBVals(3);

end

%Plot low resolution image of entire tissue slide with object locations and
%save
montage(Cat_img_array, 'Size', Mosaic_Dim, 'BackgroundColor',
'black','BorderSize',1);
savefig([cd '\' DataSetName '\Grid_Overview.fig' ]);
save([cd '\' DataSetName '\Total_objects_' DataSetName '.mat'],
'Total_Objects')
save([cd '\' DataSetName '\Ziel_objects_pre-curated' DataSetName '.mat'],
'Total_Ziel_Objects','TileIndex','Tileflag' )
```

8.2.4 Mtb object validator

Mandatory user inputs

```
%Validate individual objects in the context of the image tile in which they
%were found. This script should be run directly after refining object
%distribution with all variables still loaded in the MATLAB workspace after
%processing a full tissue slide

%Add file name of target slide
File_name = 'Slide name string.tif';

%Name of dataset to be saved after curation
DataSetName = 'Dataset name_refined';
mkdir(DataSetName);

%Change size of circles indicating Mtb positions, if necessary.
circSize = 300;
```

Loads MATLAB images and matching datasets containing Mtb objects for user inspection and individual validation

```
%Add whole tile numbers that contain no Mtb bacilli to the “Tile_list” variable
Tile_list = [];
RM3 = [];
GM3 = [];
BM3 = [];

%Plot all objects found in image tiles identified in “Tile_List” , select using
MATLAB brush tool and then export to new variable “Mtb_index2”
for kk = 1:numel(Tile_list)
   SubStruct2 = Total_Ziel_Objects([Total_Ziel_Objects.Tile_number] ==
Tile_list(kk));
   RM3 = [RM3;[SubStruct2.RGB_RedMean]'];
   GM3 = [GM3;[SubStruct2.RGB_GreenMean]'];
   BM3 = [BM3;[SubStruct2.RGB_BlueMean]'];
```

```
end
scatter3(RM3,GM3,BM3,2,'filled','r')
xlabel('Red')
ylabel('Green')
zlabel('Blue')



%%Remove objects matching those selected in Mtb_index2 above from
%%Total_Ziel_Objects structure. Only run this code with a complete
%%Tile_list vector
catRGB_ID2 = [RM2, GM2, BM2, ID2];
RemoveVect = [];
for ii = 1: size(catRGB_ID2,1)
    ii/size(catRGB_ID2,1)*100
        for jj = 1:(size(Mtb_index2,1))
            if isequal(catRGB_ID2(ii,1), Mtb_index2(jj,1)) &&
isequal(catRGB_ID2(ii,2), Mtb_index2(jj,2)) && isequal(catRGB_ID2(ii,3),
Mtb_index2(jj,3))
                RemoveVect = [RemoveVect; catRGB_ID2(ii,4)];
            else
            end
        end
end

Total_Ziel_Objects(RemoveVect) = [];
```

Identify and remove individual objects from Mtb image tiles

```
%Open a figure for an image tile and generate a list of objects to be
%excluded from the Total_Ziel_Objects dataset.
```

```
SelectROIVect = [];


%If there are fewer ZN-Mtb objects than objects that are Not ZN-Mtb in the
%figure for a tile, use the code below.

%Delete all objects that are ZN-Mtb in the active figure and then run the
%code below to add those remaining to a list of objects to exclude
h = gcf;
i = h.CurrentAxes;
j = i.Children;

    for ii = 1:(numel(j)-1);
    SelectROIVect = [SelectROIVect;str2num(j(ii).Label)];
    end

Object_exclusion_list = [SelectROIVect];

%If there are more ZN-Mtb objects than objects that are not ZN-Mtb in the
%figure for a tile, use the code below

%Run this code before deleting any objects from the the figure for a tile
h = gcf;
i = h.CurrentAxes;
j = i.Children;
TileVect = [];
for ii = 1:(numel(j)-1);
TileVect = [TileVect;str2num(j(ii).Label)];
end

%Delete all objects that are not ZN-Mtb from the figure and then run the
%code below
h = gcf;
```

```
i = h.CurrentAxes;
j = i.Children;
PosVect = [];
for ii = 1:(numel(j)-1);
PosVect = [PosVect;str2num(j(ii).Label)];
end

[idxlog idxElement] = ismember(PosVect,TileVect);
TileVect(idxElement) = [];
negVect = TileVect;
SelectROIVect = [SelectROIVect;negVect];
Object_exclusion_list = [SelectROIVect];



%Remove individual objects that have been identified as "Not ZN-Mtb" from
%the Total_Ziel_Objects dataset. Note: Only run this command once you
%have identified all the individual objects to remove from the whole
%database.
RM4 = [Total_Ziel_Objects.RGB_RedMean]';
GM4 = [Total_Ziel_Objects.RGB_GreenMean]';
BM4 = [Total_Ziel_Objects.RGB_BlueMean]';
ID4 = (1:numel(Total_Ziel_Objects))';
IDN4 = [Total_Ziel_Objects.Object_ID]';

catRGB_ID4 = [RM4, GM4, BM4, ID4, IDN4];
Obj_Rmv_Vect = [];

for ii = 1: size(catRGB_ID4,1)
    ii/size(catRGB_ID4,1)*100
        for jj = 1:(size(Object_exclusion_list,1))
            if isequal(catRGB_ID4(ii,5), Object_exclusion_list(jj,1))
                Obj_Rmv_Vect = [Obj_Rmv_Vect; catRGB_ID4(ii,4)];
            else
```

```
      end
    end
end

Total_Ziel_Objects(Obj_Rmv_Vect) = [];
```

Label validated Ziehl Neelsen stained Mtb bacilli in the whole slide database

```
ID5 = [Total_Ziel_Objects.Tag];

%%Clear existing labels
for ii = 1:numel(Total_Objects)
Total_Objects(ii).Ziel = 0;
end

%%Add manually curated labels
for ii = 1:numel(ID5)
Total_Objects(ID5(ii)).Ziel = 1;
end

%%%Export final data for convolutional neural network training
save([cd '\' DataSetName '\Ziel_objects_only_' DataSetName '.mat'], 'Total_Ziel_Objects')
save([cd '\' DataSetName '\Final_objects_Labelled' DataSetName '.mat'], 'Total_Objects')
save([cd '\' DataSetName '\FinalExcluded_Tiles.mat'], 'Tile_list')
save([cd '\' DataSetName '\FinalExcluded_IDs.mat'], 'Object_exclusion_list')

PreExport = struct2table(rmfield(Total_Objects,{'Centroid','Tile_number','Object_ID','BoundingBox'}));
```

```
Export = table2array(PreExport);
writematrix(Export,[cd '\' DataSetName '\' DataSetName],'Delimiter','\t')
```

Plot final fully curated figures

```
%%find unique tile numbers with ziel+ objects in them
ObjTiles = unique([Total_Ziel_Objects.Tile_number]);

%Identify locations of Mtb singles and aggregates and plot on whole slide
image
for ii = 1:size(TileIndex,1)

    A =
imread(File_name,'PixelRegion',{[TileIndex(ii,1),TileIndex(ii,2)],[TileIndex(ii,3)
,TileIndex(ii,4)]});

        if (sum(ii == ObjTiles)) == 1

            SinBin = zeros(size(A,1),size(A,2));
            AggBin = zeros(size(A,1),size(A,2));
            one_res_level = A;
            redLayer = one_res_level(:,:,1);
            greenLayer = one_res_level(:,:,2);
            blueLayer = one_res_level(:,:,3);

%Find single Mtb locations
            SubStruct = Total_Ziel_Objects([Total_Ziel_Objects.Tile_number] ==
ii);
            SubS = SubStruct([SubStruct.Sqr_microns] < Mtb_sz_val);
            a = [SubS.Centroid];
            b = a(1:2:end);
            c = a(2:2:end);
            d = [b',c'];

                for jj = 1:size(d,1)
                    SinBin(round(d(jj,2)),round(d(jj,1))) = 1;
                end

            SinBin = bwdist(SinBin);
            SinBin(SinBin <= circSize) = 1;
            SinBin(SinBin > circSize) = 0;
            SinBin = logical(SinBin);

%Find Mtb aggregates locations
            SubA = SubStruct([SubStruct.Sqr_microns] >= Mtb_sz_val);
            e = [SubA.Centroid];
            f = e(1:2:end);
            g = e(2:2:end);
            h = [f',g'];

                for jj = 1:size(h,1)
                    AggBin(round(h(jj,2)),round(h(jj,1))) = 1;
                end

            AggBin = bwdist(AggBin);
            AggBin(AggBin <= circSize) = 1;
            AggBin(AggBin > circSize) = 0;
            AggBin = logical(AggBin);

%Add images of Mtb positions to montage array
            redLayer(SinBin) = 255;
            greenLayer(SinBin) = 127;
            blueLayer(SinBin) = 0;

            redLayer(AggBin) = 255;
```

```
        greenLayer(AggBin) = 0;
        blueLayer(AggBin) = 127;

        RGB_4Array = cat(3,redLayer,greenLayer,blueLayer);
        Tile_Rsz = imresize(RGB_4Array,0.2);
        Tile_RszBCK = imresize(one_res_level,0.2);
        Tile_Rsz =
insertText(Tile_Rsz,[100,100],ii,'Fontsize',100,'BoxOpacity',0,'TextColor','blac
k');
        Cat_img_array{ii} = Tile_Rsz;
        Cat_img_arrayBCK{ii} = Tile_RszBCK;

        bshow = imshow(A);
        hold on
           for kk = 1:numel(SubStruct)
            h = drawcircle('Center',SubStruct(kk).Centroid,...

'Radius',30,'Label',num2str(SubStruct(kk).Object_ID),'LabelVisible','hover',...
              'color', SubStruct(kk).Size_class, 'FaceAlpha', 0,...
              'InteractionsAllowed','reshape', 'LineWidth', 4,...
              'Deletable',true);
           end
        save([cd '\' DataSetName '\FinalCentroids_Mtb_' File_name(1:end-4)
'_Tile_' num2str(TileIndex(ii,5)) '.mat'], 'SubStruct');
        savefig([cd '\' DataSetName '\FinalFigure_' File_name(1:end-4)
'_Tile_' num2str(TileIndex(ii,5)) '.fig' ]);
        close(gcf)

      else
        A =
imread(File_name,'PixelRegion',{[TileIndex(ii,1),TileIndex(ii,2)],[TileIndex(ii,3)
,TileIndex(ii,4)]});
        one_res_level = A;
```

```
            Tile_Rsz = imresize(one_res_level,0.2);
            Tile_RszBCK = imresize(one_res_level,0.2);
            Tile_Rsz = insertText(Tile_Rsz,[100,100],ii,'Fontsize',100,'BoxOpacity',0,'TextColor','black');
            Cat_img_array{ii} = Tile_Rsz;
            Cat_img_arrayBCK{ii} = Tile_RszBCK;
        end
end
%Create whole slide image and save
montage(Cat_img_array, 'Size', Mosaic_Dim, 'BackgroundColor', 'black','BorderSize',1);
savefig([cd '\' DataSetName '\Grid_OverviewFINAL.fig' ]);
toc
```

8.2.5 HyRoNet trainer

Mandatory user inputs

```
%Use data extracted and manually labelled to train a HyRoNet model to
%detect Ziel Neelson stained Mtb in tissue slides

%Add filename of training matrix file, or load separate matrices and
%concatenate to a single training dataset
Data_filename = ['Curated_training_data_matrix.txt'];

%Specify the train/validate/test split percentages
Data_split_vector = [80,20,0];

%Identify labelling column in training data matrix
Label_Column_Number = 27;
```

```
%Identify training feature columns in training data matrix
Feature_Train_Range = [2:26,28:30];

%Base number of iterations for gradient descent in network neurons
IterL1 = 50000;

%Number of hidden layers per neuron of hyronet
NLay = 1;

%Number of nodes per hidden layer of the network
NNod = numel(Feature_Train_Range)+1;

%Size of steps during gradient descent for neurons 1,2,3 and 4
alphaN1 = 0.5;
alphaN2 = 0.5;
alphaN3 = 0.5;

%Define the range of values over which to test values for lambda and the
%size of the increments between these values
Lambda_Range = [40,90];
Lambda_Resolution = 10;

%Target sensitivity for each neuron
Se1 = 98;
Se2 = 99;
Se3 = 95;
```

Load and split training data into train/validate/test subsets

```
Data = readmatrix(Data_filename);
ParamsMatrix = {};
```

```
ThreshVect = [];

%Add row number indices to end of matrix
Data = [Data,[1:size(Data,1)]'];

%split data
TVT = TVTDataSplitter(Data,Data_split_vector);

%label new variables of split data
Train = TVT{1};
Validate = TVT{2};

%Create symmetrical training dataset (where number of positive training
%examples = number of negative training examples)
Xtr = SymDataPN(Train,Label_Column_Number);
Ytr = Xtr(:,Label_Column_Number);

%Remove label column prior to training and normalize features
Xtr = Xtr(:,Feature_Train_Range);
Xtr = NormFeat(Xtr);

%Create cross validation set, remove label column and normalize features
as
%with training set
Xcv = Validate(:,Feature_Train_Range);
Xcv = NormFeat(Xcv);
Ycv = Validate(:,Label_Column_Number);

%Create full dataset with row labels to generate predictions using neuron 1
XAll = NormFeat(Data(:,Feature_Train_Range));
YAll = Data(:,Label_Column_Number);
```

Plot Learning curves

```
PlotLearningCurves(Xtr, Xcv, Ytr, Ycv, 0, YlogicArray(Ytr), 0.5)
```

Train first Neuron and get optimal regularization parameter

```
%Get parameter unregularized parameter matrix
[TParams] = Train_GradDescNN(Xtr, Ytr, IterL1, NLay, NNod, alphaN1);

%Find optimal regularization parameter to ensure good model generalization
%to cross validation dataset
optimlambda = GetLambda(Xtr, Xcv, Ytr, Ycv, Lambda_Range,
Lambda_Resolution, 0.5, IterL1, NLay, NNod, alphaN1);

%Train new CNN model with optimal regularization parameter applied
[TParams] = Train_GradDescNN(Xtr, Ytr, IterL1, NLay, NNod, alphaN1,
optimlambda);
SSPNVect = [];
for ii = 0:0.01:1

    [Se, Sp, PPV, NPV] = SSPN_NN(TParams,Xcv,Ycv,ii,0);
    SSPNVect = [SSPNVect;Se, Sp, PPV, NPV, ii];

end

%%Find prediction threshold that ensures a sensitivity closest to Se1%
[val,idx] = min(abs(SSPNVect(:,1)-Se1));
Thresh = SSPNVect(idx,5);
[Se, Sp, PPV, NPV, Predictions] = SSPN_NN(TParams,XAll,YAll,Thresh,1);

%Save model parameters and prediction threshold
```

```
ParamsMatrix{1} = {TParams};
ThreshVect = [ThreshVect;Thresh];
```

Train Neuron 2

```
[Params1, Thresh1, SubsetIDs1] = TrainHyRoParams(Data, Predictions,
IterL1*3, NLay, NNod, alphaN2, Se2);

ParamsMatrix{2} = {Params1};
ThreshVect = [ThreshVect,Thresh1];

Predictions1  = PredictionLabels(SubsetIDs1,Data);
```

Train Neuron 3

```
[Params2, Thresh2, SubsetIDs2] = TrainHyRoParamsUnSym(Data,
Predictions1, IterL1*4, NLay, NNod, alphaN3, Se3);

ParamsMatrix{3} = {Params2};
ThreshVect = [ThreshVect,Thresh2];

Predictions2  = PredictionLabels(SubsetIDs2,Data);
```

Save model parameters, predictions and thresholds

```
AllPredictions = [Predictions,Predictions1,Predictions2];

save([cd '\Model_Parameter_matrices'],'ParamsMatrix')
save([cd '\All_predictions'],'AllPredictions')
save([cd '\ThreshVect'],'ThreshVect')
```

8.2.6 TVT Data splitter

```
function [TVT] = TVTDataSplitter(X,SplitVect)

%Note: X must be a m x n matrix, where m is the number of observations
%and n is the number of features. Splitvect is a vector describing the desired
%data split. e.g [60 20 20] would return 60% of the data in the training
%set, 20% in the validation set and the final 20% in the test set (note,
%the data split must add up to 100%)

if ~exist('SplitVect', 'var') || isempty(SplitVect)
    SplitVect = [60,20,20];
end

Rando = randperm(size(X,1));
SplitCell = cell(1,size(SplitVect,2));

for ii = 1:numel(SplitVect)

    split = floor((SplitVect(ii)/100)*size(X,1));
    SplitCell{ii} = X(Rando(1:split),:);
    Rando(:,(1:split)) = [];

    if ii == numel(SplitVect)
        SplitCell{1} = [SplitCell{1}; X(Rando(:),:)];
    end

end

TVT = SplitCell;

end
```

8.2.7 Dataset symmetrizer

```
function [SymMat] = SymDataPN(DataMatrix,LabelColumnNumber)
%%Takes a training set with many fewer positive examples than negatives
%and creates a new symmetrical matrix containing as many positive
%examples as negatives in a randomized order

    Pos = DataMatrix(DataMatrix(:,LabelColumnNumber)==1,:);
    Neg = DataMatrix(DataMatrix(:,LabelColumnNumber)==0,:);
    k = randperm(size(Neg,1));
    NewNeg = Neg(k(1:size(Pos,1)),:);
    NRSymMat = [Pos;NewNeg];
    SymMat = NRSymMat(randperm(size(NRSymMat, 1)),:);

end
```

8.2.8 Feature Normalizer

```
function [NormFeat] = NormFeat(X)
%Normalize all features to be within a similar scale.

%Concatenate rows to add range of background RGB values derived from
%training data. Accounts for between slide variability
NormLimsMin = [ nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan,
nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, 63,
104, 105 ];
NormLimsMax = [ nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan,
nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, 149,
180, 133];
```

```
    X = [X;NormLimsMin;NormLimsMax];

        for ii = 1:size(X,2)
            X(1:(end-2),ii) = (X(1:(end-2),ii) - mean(X(1:(end-2),ii)))./(max(X(:,ii))-min(X(:,ii)));
        end
        X = X(1:(end-2),:);

NormFeat = X;
```

8.2.9 CNN gradient descent

```
function [TParams, Cost_History] = Train_GradDescNN(X, Y, num_iters, HiddenLayers, HiddenlayerNodes, alpha, lambda, PredThresh, Klabels)
%%Note: X must be an m x n vector. Where m is the number of training
%%examples and n is the number of features.

%%Set lambda to zero if not provided
if ~exist('PredThresh', 'var') || isempty(PredThresh)
    PredThresh = 0.5;
end

%%Set lambda to zero if not provided
if ~exist('lambda', 'var') || isempty(lambda)
    lambda = 0;
end

%%Get number of unique labels (output classes)
if ~exist('Klabels', 'var') || isempty(Klabels)
    [~, Klabels] = YlogicArray(Y);
else
```

```
    [~, Klabels] = YlogicArray(Y,Klabels);
end

%%Randomly initialize weights
rng(1);
InitWeights =
Initialize_NNweights(size(X,2),HiddenLayers,HiddenlayerNodes,Klabels);
Params = InitWeights;
cost = zeros(1,num_iters);
%%Get model gradients, cost and predictions
for iter = 1:num_iters
    [~, Activations]= Forwardprop(Params,X);
    Grad = Backprop(Activations,Params,X,Y, lambda,Klabels);

    for ii = 1:numel(Params)
    Params{ii} = Params{ii} - alpha*Grad{ii}';
    end

    cost(iter) = CostFunctionNN(Params,X,Y,lambda,Klabels);
end

%%Plot cost and percentage correct classification for newly trained
%%parameters
Cost_History = cost;
TParams = Params;
[H_O, ~]= Forwardprop(TParams,X);

if size(H_O,1) > 1
    [~,Predictions] = max(H_O,[],1);
    PercPred = mean(double(Predictions' == Y))*100;
else
    Predictions = H_O;
    Predictions(Predictions >= PredThresh) = 1;
    Predictions(Predictions < PredThresh) = 0;
    PercPred = mean(double(Predictions' == Y))*100;
end
figure;
plot(Cost_History, 'Color','#0072BD', 'LineWidth',4)
title(['Training set accuracy : ' num2str(PercPred) '%'],'FontSize', 20)
xlabel('Iteration','FontSize', 20)
ylabel('Cost (J(theta))','FontSize', 20)
set(gcf,'position',[500,300,1000,600])
set(gca,'FontSize', 20)
end
```

8.2.10 CNN Lambda calculator

```
function optimlambda = GetLambda(Xtrain, Xcv, Ytrain, Ycv, range,
resolution, PredThresh, num_iters, HiddenLayers, HiddenLayerNodes,
alpha)
%Iteratively runs a full CNN training pipeline for different values of Lambda,
applies the resultant model to training and cross validation datsets and plots
the cost results.
if ~exist('PredThresh', 'var') || isempty(PredThresh)
    PredThresh = 0.5;
end

if ~exist('range', 'var') || isempty(range)
    range = [0 , 10000];
end

if ~exist('resolution', 'var') || isempty(resolution)
    resolution = 10;
end
```

```
lambda = linspace(range(1),range(2),resolution);
CostT = zeros(size(lambda));
CostCV = zeros(size(lambda));
SSPNvect = zeros(4,size(lambda,2));

for ii = 1:numel(lambda)

   TParams = Train_GradDescNN(Xtrain, Ytrain, num_iters, HiddenLayers,
HiddenLayerNodes, alpha, lambda(ii));
   CostT(ii) = (CostFunctionNN(TParams,Xtrain,Ytrain));
   CostCV(ii) = (CostFunctionNN(TParams,Xcv,Ycv));
   [Se, Sp, PPV, NPV] = SSPN_NN(TParams,Xcv,Ycv,PredThresh,0);
   SSPNvect(:,ii) = [Se; Sp; PPV; NPV];

end
optimlambdaind = CostCV == min(CostCV);
optimlambda = lambda(find(optimlambdaind,1));

figure;
plot(lambda,CostT,'Color','#0072BD', 'LineWidth',4)
hold on
plot (lambda,CostCV,'Color','#D95319', 'LineWidth',4)
title(['Optimal lambda: ' num2str(optimlambda)],'FontSize', 20)
legend('Training set','Cross validation set','FontSize',
20,'Location','southeast')
xlabel('Lambda')
ylabel('Cost')
set(gca,'FontSize',20)

figure
plot(lambda,SSPNvect(1,:), 'LineWidth',2)
hold on
```

```
plot(lambda,SSPNvect(2,:), 'LineWidth',2)
hold on
plot(lambda,SSPNvect(3,:), 'LineWidth',2)
hold on
plot(lambda,SSPNvect(4,:), 'LineWidth',2)
legend('Sensitivity','Specificity','+ Predictive val', '- Predictive val','FontSize',
20,'Location','southeast')
xlabel('Lambda','FontSize', 20)
ylabel('Percent','FontSize', 20)
title('SSPN vs lambda','FontSize', 20)
set(gca,'FontSize',20)
end
```

8.2.11 CNN performance calculator

```
function [Se, Sp, PPV, NPV,Predictions] =
SSPN_NN(TrainedParams,X,Y,PredThresh,plotting)
%Calculates sensitivity, specificity, positive predive value, negative predictive
values and generates predictions using trained model
%parameters

if ~exist('plotting', 'var') || isempty(plotting)
    plotting = 1;
end

if ~exist('PredThresh', 'var') || isempty(PredThresh)
    PredThresh = 0.5;
end



H_O = Forwardprop(TrainedParams,X);
```

```
  if size(H_O,1) > 1
    [~,Predictions] = max(H_O,[],1);
    PercPred = mean(double(Predictions' == Y))*100;
  else
    Predictions = H_O;
    Predictions(Predictions >= PredThresh) = 1;
    Predictions(Predictions < PredThresh) = 0;
    PercPred = mean(double(Predictions' == Y))*100;
  end

Predictions = Predictions';
SSPN = [Y,Predictions];

Se = sum(SSPN(:,1)) / (sum(SSPN(:,1)) + (sum(SSPN(:,1) == 1 & SSPN(:,2) == 0))) *100;
Sp = sum(SSPN(:,1)==0) / (sum(SSPN(:,1)==0) + (sum(SSPN(:,1) == 0 & SSPN(:,2) == 1)))*100;
PPV = sum(SSPN(:,1)) / (sum(SSPN(:,1)) + (sum(SSPN(:,1) == 0 & SSPN(:,2) == 1)))*100;
NPV = sum(SSPN(:,1)==0) / ((sum(SSPN(:,1) == 1 & SSPN(:,2) == 0)) + sum(SSPN(:,1)==0))*100;
%Plot performance metrics
if plotting == 1
  figure;
  xvalues = {'Sensitivity','Specificity','+ Predictive val', '- Predictive val'};
  yvalues = {'Percent'};
  g = heatmap(xvalues,yvalues,[Se, Sp, PPV, NPV])
  colormap(winter)
  title('SSPN')
  xlabel('')
  ylabel('')
  caxis([0, 100]);
```

```
    g.FontSize = 20;

    figure;
    xvalues = {'True','False'};
    yvalues = {'Predict True', 'Predict False'};
    h = heatmap(xvalues,yvalues,[(sum(SSPN(:,1) == 1 & SSPN(:,2) ==
1))/size(SSPN,1)*100, (sum(SSPN(:,1) == 0 & SSPN(:,2) ==
1))/size(SSPN,1)*100; (sum(SSPN(:,1) == 1 & SSPN(:,2) ==
0))/size(SSPN,1)*100, (sum(SSPN(:,1) == 0 & SSPN(:,2) ==
0))/size(SSPN,1)*100])
    colormap(winter)
    title('Percentage data composition')
    h.FontSize = 20;

    PredThreshVect = 0:0.01:1;
    SSPNvect = zeros(4,size(PredThreshVect,2));

    for ii = 1:numel(PredThreshVect)

        Predictions2 = H_O;
        Predictions2(Predictions2 >= PredThreshVect(ii)) = 1;
        Predictions2(Predictions2 < PredThreshVect(ii)) = 0;
        Predictions2 = Predictions2';
        SSPN2 = [Y,Predictions2];

        Se2 = sum(SSPN2(:,1)) / (sum(SSPN2(:,1)) + (sum(SSPN2(:,1) == 1 &
SSPN2(:,2) == 0))) *100;
        Sp2 = sum(SSPN2(:,1)==0) / (sum(SSPN2(:,1)==0) + (sum(SSPN2(:,1)
== 0 & SSPN2(:,2) == 1)))*100;
        PPV2 = sum(SSPN2(:,1)) / (sum(SSPN2(:,1)) + (sum(SSPN2(:,1) == 0
& SSPN2(:,2) == 1)))*100;
        NPV2 = sum(SSPN2(:,1)==0) / ((sum(SSPN2(:,1) == 1 & SSPN2(:,2) ==
0)) + sum(SSPN2(:,1)==0))*100;
```

```
        SSPNvect(:,ii) = [Se2; Sp2; PPV2; NPV2];

    end

    figure
    plot(PredThreshVect,SSPNvect(1,:), 'LineWidth',4)
    hold on
    plot(PredThreshVect,SSPNvect(2,:), 'LineWidth',4)
    hold on
    plot(PredThreshVect,SSPNvect(3,:), 'LineWidth',4)
    hold on
    plot(PredThreshVect,SSPNvect(4,:), 'LineWidth',4)
    legend('Sensitivity','Specificity','+ Predictive val', '- Predictive val',
'FontSize', 20, 'Location','southeast')
    xlabel('Prediction Threshold', 'FontSize', 20)
    ylabel('Percent', 'FontSize', 20)
    title('SSPN vs Prediction Threshold', 'FontSize', 20)
    set(gca,'FontSize',20)

else

end

end
```

8.2.12 HyRoNet neuron trainer

```
function [Params, Thresh, SubsetIDs] =
TrainHyRoParams(Training_Set_with_RowIDs, Predictions, num_iters,
HiddenLayers, HiddenLayerNodes, alpha, sensitivity,TrainCols,LabelCol)
```

```
%Trains HyRoNet neuron, as called in the TrainGradDesc_CNN function,
%and returns predictions to apply to original input data with corresponding
%thresholds to achive specified sensitivity

TrainCols = TrainCols-1;
LabelCol = LabelCol-1;
Training_Set_with_RowIDs = Training_Set_with_RowIDs(:,2:end);
Training_Set_with_RowIDs(:,TrainCols) =
NormFeat(Training_Set_with_RowIDs(:,TrainCols));
Training_Set2 = [Training_Set_with_RowIDs,Predictions];
Training_Set2B = Training_Set2([Training_Set2(:,end) == 1],(1:(end-1)));

%Split data into train and validate set
TVT = TVTDataSplitter(Training_Set2B,[80,20,0]);

Xtr = TVT{1};
Xtr = SymDataPN(Xtr,LabelCol);
Xcv = TVT{2};
Ytr = Xtr(:,LabelCol);
Ycv = Xcv(:,LabelCol);
Xtr = Xtr(:,TrainCols);
Xcv = Xcv(:,TrainCols);

%Create data matrix to apply predictions
XAll = [TVT{1};TVT{2}];
YAll = XAll(:,LabelCol);
XAll = XAll(:,[TrainCols,(TrainCols(end)+1)]);

%Train network neuron
[TParams, Cost_History] = Train_GradDescNN(Xtr, Ytr, num_iters,
HiddenLayers, HiddenLayerNodes, alpha);

TParams1 = TParams;
```

```
SSPNVect = [];

for ii = 0:0.01:1

    [Se, Sp, PPV, NPV] = SSPN_NN(TParams1,Xcv,Ycv,ii,0);
    SSPNVect = [SSPNVect;Se, Sp, PPV, NPV, ii];

end

%%Find prediction threshold that matches target sensitivity
[val,idx] = min(abs(SSPNVect(:,1)-sensitivity));
Thresh = SSPNVect(idx,5);
[Se_, Sp_, PPV_, NPV_, NewPred] = SSPN_NN(TParams1,XAll(:,[1:(end-1)]),YAll,Thresh,1);

XAllLabelled = [XAll,NewPred];
SubsetIDs = XAllLabelled(XAllLabelled(:,end)==1,end-1);

Params = TParams1;
end
```

8.2.13 HyRoNet Mtb classifier

```
function [predictions, full_predictions] = HyroNetGetMtb(Extracted_Data)
%Generate predictions for data using parameters generated during HyRoNet
%training

load([cd 'Model_Parameter_matrices.mat']);
load([cd 'ThreshVect.mat']);

Params1 = ParamsMatrix{1,1};
```

```
Params1 = Params1{1,1};

Params2 = ParamsMatrix{1,2};
Params2 = Params2{1,1};

Params3 = ParamsMatrix{1,3};
Params3 = Params3{1,1};

Thresh1 = ThreshVect(1);
Thresh2 = ThreshVect(2);
Thresh3 = ThreshVect(3);

Extracted_Data = [Extracted_Data,[1:size(Extracted_Data,1)]'];
Extracted_Data = Extracted_Data(:,2:end);
Extracted_Data(:,[1:(end-1)]) = NormFeat(Extracted_Data(:,[1:(end-1)]));
```

Generate predictions for neuron 1

```
H_O1 = Forwardprop(Params1,Extracted_Data(:,[1:(end-1)]));
    if size(H_O1,1) > 1
        [~,Predictions] = max(H_O1,[],1);
    else
        Predictions = H_O1;
        Predictions(Predictions >= Thresh1) = 1;
        Predictions(Predictions < Thresh1) = 0;
    end
H_O1 = Predictions;

Extracted_Data2 = [Extracted_Data,H_O1'];
SubsetIDs1 = Extracted_Data2(Extracted_Data2(:,end)==1,end-1);
Predictions1  = PredictionLabels(SubsetIDs1,Extracted_Data);
Extracted_Data2 = [Extracted_Data,Predictions1];
```

```
Extracted_Data2 = Extracted_Data2([Extracted_Data2(:,end) == 1],(1:(end-1)));
```

Generate predictions for neuron 2

```
H_O2 = Forwardprop(Params2,Extracted_Data2(:,[1:(end-1)]));
   if size(H_O2,1) > 1
      [~,Predictions] = max(H_O2,[],1);
   else
      Predictions = H_O2;
      Predictions(Predictions >= Thresh2) = 1;
      Predictions(Predictions < Thresh2) = 0;
   end
H_O2 = Predictions;


Extracted_Data3 = [Extracted_Data2,H_O2'];
SubsetIDs2 = Extracted_Data3(Extracted_Data3(:,end)==1,end-1);
Predictions2  = PredictionLabels(SubsetIDs2,Extracted_Data);
Extracted_Data3 = [Extracted_Data,Predictions2];
Extracted_Data3 = Extracted_Data3([Extracted_Data3(:,end) == 1],(1:(end-1)));
```

Generate predictions for neuron 3

```
H_O3 = Forwardprop(Params3,Extracted_Data3(:,[1:(end-1)]));
   if size(H_O3,1) > 1
      [~,Predictions] = max(H_O3,[],1);
   else
      Predictions = H_O3;
      Predictions(Predictions >= Thresh3) = 1;
      Predictions(Predictions < Thresh3) = 0;
   end
H_O3 = Predictions;
```

```
Extracted_Data4 = [Extracted_Data3,H_O3'];
SubsetIDs3 = Extracted_Data4(Extracted_Data4(:,end)==1,end-1);
Predictions3  = PredictionLabels(SubsetIDs3,Extracted_Data);
Extracted_Data4 = [Extracted_Data,Predictions3];
Extracted_Data4 = Extracted_Data4([Extracted_Data4(:,end) == 1],(1:(end-1)));

full_predictions = [Predictions1,Predictions2,Predictions3];
predictions = Predictions3;
end
```

8.2.14 CNN Weight initializer

```
function Params = Initialize_NNweights(InputLNodes,NHiddenL,HiddenLNodes,OutputLNodes)
%Note that this function assumes all hidden layers have the same number of
%nodes in each layer. And adds weights for the bias unit. X must be an m x
%n matrix, where m is the number of training examples and n is the number
%of features

TLayers = NHiddenL + 2;

Params = cell(1,TLayers-1);

for ii = 1:TLayers-1

    if ii == 1

        epsilon_init = (sqrt(6))/(sqrt(InputLNodes + HiddenLNodes));
        Params{ii} = rand(InputLNodes+1, HiddenLNodes) * 2 * epsilon_init -
```

```
epsilon_init;

    elseif ii == (TLayers-1)

        epsilon_init = (sqrt(6))/(sqrt(HiddenLNodes + OutputLNodes));
        Params{ii} = rand(HiddenLNodes+1, OutputLNodes) * 2 * epsilon_init -
epsilon_init;

    else

        epsilon_init = (sqrt(6))/(sqrt(HiddenLNodes + HiddenLNodes));
        Params{ii} = rand(HiddenLNodes+1, HiddenLNodes) * 2 * epsilon_init -
epsilon_init;

    end

end

end
```

8.2.15 CNN Forward propagator

```
function [H_O, Activations]= Forwardprop(Params,Features)
%Notes: Features should be an m x n matrix, where m is the number of
%training examples and n is the number of features.

%%calculate number of layers
NumLayers = size(Params,2)+1;

%%Add bias unit to features
X = [ones(size(Features,1),1),Features]';
```

```
%%Create cell array for node activations
Z = cell(1,NumLayers-1);
%%Calculate activations
for ii = 1: NumLayers-1

    if ii == 1
        Z{ii} = Params{ii}'*X;
        Z{ii} = sigmoid(Z{ii});
        Z{ii} = [ones(1,size(Z{ii},2));Z{ii}];

    elseif ii == NumLayers-1
        Z{ii} = Params{ii}'*Z{ii-1};
        Z{ii} = sigmoid(Z{ii});
        H_O = Z{ii};

    else
        Z{ii} = Params{ii}'*Z{ii-1};
        Z{ii} = sigmoid(Z{ii});
        Z{ii} = [ones(1,size(Z{ii},2));Z{ii}];

    end

end

Activations = Z;

end
```

8.2.16 CNN Backpropagator

```
function Grad = Backprop(Activations,Params,Features,Y,lambda,Klabels)
%%If lambda is not supplied, sets lambda to zero such that the gradients
```

```
%%returned will be unregularized
if ~exist('lambda', 'var') || isempty(lambda)
    lambda = 0;
end

%%add bias term to feature set
X = [ones(size(Features,1),1),Features];

%%convert y ground truths vector to multiple hypothesis binary matrix if
%%necessary
if ~exist('Klabels', 'var') || isempty(Klabels)
    [y_logical, ~] = YlogicArray(Y);
    Y = y_logical';
else
    [y_logical, ~] = YlogicArray(Y,Klabels);
    Y = y_logical';
end

%%create cell arrays for delta values
Delta = cell(1,size(Activations,2));
Delta_ijl = cell(1,size(Params,2));

%%backpropagation to calculate final deltas used to calculate gradient term
for ii = numel(Activations):-1:1

    if ii == numel(Activations)
        Delta{ii} = Activations{ii} - Y;
        Delta_ijl{ii} =  Delta{ii}*Activations{ii-1}';
    else

        Delta{ii} = ((Params{ii+1}(2:end,:))*Delta{ii+1}) .*
(Activations{ii}(2:end,:).*(1-Activations{ii}(2:end,:))); %%
```

```
            if ii == 1
                Delta_ijl{ii} =  Delta{ii}*X;
            else
                Delta_ijl{ii} =  Delta{ii}*Activations{ii-1}';
            end
    end
end

%%preallocate and assign useful variables
Grad = cell(1,size(Delta_ijl,2));
m = size(X,1);

%%calculate gradients(regularized implimentation)
for jj = 1:numel(Delta_ijl)

    Grad{jj} = (1/m)*Delta_ijl{jj};
    Grad{jj}(:,2:end) = Grad{jj}(:,2:end) + (lambda/m)*(Params{jj}(2:end,:)');
%%Regularized gradient

end



end
```

8.2.17 CNN Cost calculator

```
function cost = CostFunctionNN(Params,Features,Y,lambda,Klabels)
%%note: Y input must be in the form of numeric classes, the lowest class
%%being 1 (zero indexing not possible in matlab). Params must be
parameter randomly initialized parameters from the "Initialize_NNweights"
function
```

```
%%If lambda is not supplied, sets lambda to zero such that the cost
%%returned will be unregularized
if ~exist('lambda', 'var') || isempty(lambda)
    lambda = 0;
end

%%Convert label vector to logical label matrix, if necessary
if ~exist('Klabels', 'var') || isempty(Klabels)
    [y_logical, Klabels] = YlogicArray(Y);
else
    [y_logical, Klabels] = YlogicArray(Y,Klabels);
end

%%Generate preictions and useful variables
H_O = Forwardprop(Params, Features)';

%%Separate traning examples into each of their label classes and calculate
%%cost for each of the k classes
Yk = cell(1,Klabels);
H_Ok = cell(1,Klabels);
Jk = zeros(1,Klabels);

for jj = 1:Klabels

    if Klabels == 1
        Yk{jj} = y_logical;
        H_Ok{jj} = H_O;
        Jk(jj) = sum(sum((-Yk{jj}.*log(H_Ok{jj}) - ((1 - Yk{jj}).*log(1 -
H_Ok{jj})))));

    else

        Yk{jj} = y_logical((Y(:,1)==jj),:);
        H_Ok{jj} = H_O((Y(:,1)==jj),:);
        Jk(jj) = sum(sum((-Yk{jj}.*log(H_Ok{jj}) - ((1 - Yk{jj}).*log(1 - H_Ok{jj})))));

    end

end

%%calculate total unregularized cost
m = size(Features,1);
URcost = ((1/m)*(sum(Jk)));
%%calculate regularization terms
RegVect = zeros(1,numel(Params));

for kk = 1:numel(Params)

    RegVect(kk) = sum(sum(Params{kk}(2:end,:).^2));

end

%%Calculate final regularized cost
cost = URcost + (lambda/(2*m))*sum(RegVect);

end
```